



TRABALHO DE GRADUAÇÃO

BYEBYEFILA - SISTEMA PARA CONTROLE E AGILIDADE DE PAGAMENTO

**Ricardo Vinicius Lopes Castro
Victor Shoiti Rocha Tsujimoto**

Brasília, julho de 2018



UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Departamento de Engenharia Elétrica

TRABALHO DE GRADUAÇÃO

**BYEBYEFILA - SISTEMA PARA CONTROLE E
AGILIDADE DE PAGAMENTO**

**Ricardo Vinicius Lopes Castro
Victor Shoiti Rocha Tsujimoto**

Trabalho de Conclusão de Curso submetido ao Departamento de Engenharia
Elétrica, como parte dos requisitos necessários para obtenção do grau de
Engenheiro de Redes de Comunicação

Banca Examinadora

Prof. Georges Amvame Nze, Dr., UnB/ENE
(Orientador)

Prof. Ugo Silva Dias, Dr., UnB/ENE
(Examinador interno)

Prof. Fábio Lúcio Lopes de Mendonça, Msc., Latitude/UnB
(Examinador interno)

Dedicatória (s)

Ao meus pais, irmãs, amigos e a Jhessyka.

Ricardo Vinicius Lopes Castro

*Aos meus pais, João K. Tsujimoto e Ana
Lucia R. Tsujimoto.*

Victor Shoiti Rocha Tsujimoto

Agradecimentos

A minha mãe pelo suporte, conselhos e incentivo durante essa jornada.

Ao meu pai por ser um exemplo de dedicação e perseverança.

A minhas irmãs pela incentivo e ajuda.

A minha companheira de vida, Jhessyka, pela paciência, compreensão, incentivo e dedicação.

A todos os professores e colegas que contribuíram para esse projeto.

Ricardo Vinicius Lopes Castro

Agradeço à minha família, principalmente aos meus pais pelo incentivo e sacrifícios feitos durante minha jornada.

Agradeço aos meus amigos pelo apoio e suporte.

Agradeço aos bons professores que contribuíram para que tivesse uma boa formação acadêmica.

Victor Shoiti Rocha Tsujimoto

Resumo

Este trabalho foi motivado pelas constantes filas encontradas em estabelecimentos comerciais como bares, restaurantes e casas noturnas. Um aplicativo mobile aliado à um sistema web, pode agilizar a rotina de pagamento e liberação de saída, trazendo assim maior satisfação tanto para o cliente e quanto para o estabelecimento.

Para a implementação do sistemas web e aplicativo Android, foram estudados os fundamentos teóricos e as principais tecnologias utilizadas atualmente. Os tópicos abordados são: o servidor lamp, scriptcase, phpmyadmin; as linguagens utilizadas, PHP, HTML, CSS e JavaScript; as ferramentas de desenvolvimento Android, react-native e Android Studio; UML, Modelo entidade relacionamento e dicionário de dados.

O desenvolvimento foi feito em sete etapas, estudo sobre o funcionamento de casas noturnas, elaboração da proposta, preparação do ambiente (servidor), criação dos artefatos (Diagrama de caso de uso, modelo entidade relacionamento e Dicionário de Dados), criação do banco de dados, desenvolvimento do sistema web e por fim desenvolvimento do aplicativo Android.

Na última seção são apresentados os resultados, conclusões e trabalhos futuros.

Palavras-chave: *PHP. HTML. CSS. JavaScript. Android. UML. React-Native. Scriptcase. LAMP. phpMyAdmin. Webservice. Byebyefila. Comanda digital. Pagamento. Fila. Aplicativo.*

SUMÁRIO

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO	1
1.2	OBJETIVOS DO PROJETO	2
1.2.1	OBJETIVO GERAL	2
1.2.2	OBJETIVOS ESPECÍFICOS	2
1.3	ESTRUTURA DO TRABALHO	3
2	FUNDAMENTAÇÃO TEÓRICA.....	4
2.1	SISTEMAS WEB	4
2.2	SERVIDOR LAMP.....	5
2.2.1	LINUX	5
2.2.2	APACHE	6
2.2.3	MARIADB	7
2.2.4	PHP.....	8
2.2.5	SCRIPTCASE	9
2.2.6	PHPMYADMIN.....	9
2.3	LINGUAGENS UTILIZADAS	9
2.3.1	HTML.....	10
2.3.2	CSS.....	10
2.3.3	JAVASCRIPT.....	11
2.3.4	REACT NATIVE	11
2.4	ANDROID	12
2.4.1	ANDROID STUDIO	12
2.5	UML	13
2.5.1	DIAGRAMA DE CASO DE USO.....	14
2.6	MODELO ENTIDADE-RELACIONAMENTO.....	15
2.6.1	DICIONÁRIO DE DADOS.....	16
3	METODOLOGIA E ARTEFATOS.....	17
3.1	DELIMITAÇÃO DO TEMA	17
3.2	ESTUDO SOBRE O FUNCIONAMENTO DE CASAS NOTURNAS	18
3.3	ELABORAÇÃO DE PROPOSTA	19
3.4	CRIAÇÃO DOS ARTEFATOS	19

3.4.1	DIAGRAMA DE CASO DE USO	20
3.4.2	MODELO ENTIDADE-RELACIONAMENTO.....	44
3.4.3	DICIONÁRIO DE DADOS.....	46
4	DESENVOLVIMENTO	52
4.1	PREPARAÇÃO DO AMBIENTE	52
4.2	CRIAÇÃO DO BANCO DE DADOS.....	53
4.3	DESENVOLVIMENTO APLICAÇÃO WEB	54
4.4	DESENVOLVIMENTO APLICATIVO ANDROID.....	55
5	RESULTADOS E ANÁLISES	57
5.1	SISTEMA WEB.....	57
5.2	APLICATIVO ANDROID	61
6	CONCLUSÃO E TRABALHOS FUTUROS.....	67
	REFERÊNCIAS BIBLIOGRÁFICAS.....	69

LISTA DE FIGURAS

2.1	Estrutura de um sistema web	4
2.2	Componentes da pilha LAMP	5
2.3	O kernel estabelece a comunicação entre aplicações e recursos de hardware ...	6
2.4	Funcionamento de um servidor web	6
2.5	Estrutura de como uma aplicação acessa dados presentes no BD	7
2.6	Funções de cada linguagem, HTTP, CSS e Javascript	10
2.7	Vantagens de se utilizar o react-native	11
3.1	Diagrama de Caso de Uso	20
3.2	Modelo Entidade-Relacionamento	45
5.1	Página Inicial - Login	57
5.2	Página após o login	58
5.3	Empresa	58
5.4	Controle de cardápio e produto	59
5.5	Comanda	59
5.6	Comanda	60
5.7	Segurança	60
5.8	Tela de Login do Aplicativo	62
5.9	Campos para o cadastro de cliente	63
5.10	Menu Principal	64
5.11	Código da comanda	65
5.12	Comanda eletrônica	66

LISTA DE TABELAS

3.1	Atores dos casos de uso	21
3.2	Caso de Uso - Cadastrar Cliente	22
3.3	Caso de Uso - Pesquisar Cliente	23
3.4	Caso de Uso - Editar Cliente.....	24
3.5	Caso de Uso - Excluir Cliente	25
3.6	Caso de Uso - Cadastrar Produto	26
3.7	Caso de Uso - Editar Produto.....	27
3.8	Caso de Uso - Excluir Produto	28
3.9	Caso de Uso - Cadastrar Dados Empresa.....	29
3.10	Caso de Uso - Editar Dados Empresa	29
3.11	Caso de Uso - Cadastrar Funcionário.....	30
3.12	Caso de Uso - Pesquisar Funcionário.....	31
3.13	Caso de Uso - Editar Funcionário	32
3.14	Caso de Uso - Excluir Funcionário.....	33
3.15	Caso de Uso - Realizar Pedido	34
3.16	Caso de Uso - Editar Pedido	35
3.17	Caso de Uso - Excluir Pedido	36
3.18	Caso de Uso - Pesquisar Pedido	37
3.19	Caso de Uso - Abrir Comanda	38
3.20	Caso de Uso - Editar Comanda	39
3.21	Caso de Uso - Encerrar Comanda	40
3.22	Caso de Uso - Pesquisar Comanda	41
3.23	Caso de Uso - Visualizar Comanda	42
3.24	Caso de Uso - Imprimir Comanda.....	43
3.25	Caso de Uso - Realizar Pagamento.....	44
3.26	Tabela s_produto.....	46
3.27	Tabela s_pedido	46
3.28	Tabela s_empresa.....	47
3.29	Tabela s_usuario	48
3.30	Tabela s_comanda	49
3.31	Tabela s_usuario_s_perfil.....	49
3.32	Tabela s_perfil	49
3.33	Tabela sec_logged	50

3.34	Tabela sec_perfil_app.....	50
3.35	Tabela sec_app	51

LISTA DE TERMOS E SIGLAS

BD	Banco de Dados
CGI	Computing Graphic Image
CRUD	Create Read Update Delete
CSS	Cascading Style Sheets
FLOSS	Free Libre and Open Source Software
GNU	GNU is not Unix
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Ambiente de desenvolvimento integrado
JSON	JavaScript Object Notation
LAMP	Linux Apache Mysql PHP
MER	Modelo Entidade-Relacionamento
PHP	PHP: hypertext processor
REST	Representational Object Tranfer
SDK	Software Development Kit
SFC	Software Freedom Conservancy
SGDB	Sistema de Gerenciamento de Banco de Dados
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
UML	Unified Modeling Language
WAMP	Windows Apache Mysql PHP
XML	Extensible Markup Language

Capítulo 1

INTRODUÇÃO

Como afirma Alan Lakein [1] e [2], tempo é vida. Ele é irreversível e insubstituível. Desperdiçar o seu tempo é desperdiçar a sua vida, mas ser senhor do seu tempo é ser senhor da sua vida e fazer o melhor dela. Em um tempo onde a tecnologia cresce de forma tão acelerada, não faz sentido manter práticas antigas que consomem muito tempo. A sociedade atual vive em ritmo acelerado demandando agilidade em todas as operações do dia a dia, seja um cliente ou um estabelecimento.

Se pensarmos em como melhorar do tempo das pessoas podemos perceber que, tendo a tecnologia como aliada, é possível termos ganhos expressivos. Por exemplo, se considerarmos apenas as operações de pagamento, notamos que utilizando a tecnologia no processo de pagamento podemos fazer os clientes e estabelecimentos ganharem tempo. Para os clientes, essa agilidade representa um melhor aproveitamento de seu tempo e menos desgaste de ter que ficar em uma fila aguardando sua vez. Para o estabelecimento ou comerciante, essa agilidade representa um melhor aproveitamento do tempo de seus funcionários e atendimento a um número maior de clientes.

A contribuição da tecnologia em atividades cotidianas tende a aumentar devido ao crescimento de utilização de dispositivos moveis. Essa mobilidade permite aos usuários terem acesso a serviços, programas e poder computacional de qualquer lugar e a qualquer momento. Segundo pesquisa do site Statista [3], em 2020 teremos atingido o valor de 2,87 bilhões usuários com smartphone.

Este trabalho explora o uso da tecnologia para rever algumas práticas que podem ser aprimoradas pensando a otimização de tempo e recursos.

1.1 Motivação

A motivação deste trabalho surgiu das constantes filas que são encontradas para realizar o pagamento de comandas e liberação na saída de estabelecimentos. Com o intuito de reduzir ou até mesmo eliminar as filas em estabelecimentos, acreditamos ser possível agregar valor

ao negócio do estabelecimento incorporando tecnologia e mobilidade ao se utilizar aplicativo mobile para pagamento de comanda e mecanismo de liberação.

1.2 Objetivos do Projeto

1.2.1 Objetivo Geral

Este trabalho tem por objetivo a criação de dois sistemas: um para o estabelecimento comercial e outro para o cliente. Esses sistemas visam a otimização do tempo para os estabelecimentos comerciais e clientes. Temos como objetivos gerais:

1. Criação de um sistema web que permita ao estabelecimento criar cardápio, atender pedido do cliente, controle dos funcionários e clientes e total controle das comandas de forma digital, incluindo pagamento;
2. Desenvolvimento de um aplicativo Android que facilite o pagamento da comanda e o controle de saída dos clientes de estabelecimentos que utilizem o sistema web do item anterior. Dessa forma, reduzindo ou até eliminando a necessidade de os clientes formarem filas para as duas situações.

1.2.2 Objetivos Específicos

Para atingir os objetivos do projeto proposto, os objetivos específicos são:

- Desenvolvimento do sistema web para que o estabelecimento possa ter maior controle do seu negócio. O sistema web terá controle dos produtos, controle de funcionários, controle de pedidos e comanda digital;
- Integração do aplicativo Android com a comanda digital criada pelo sistema web;
- Fornecer o acompanhamento dos gastos aos clientes utilizando aplicativo Android para visualização de comanda;
- Facilitar o processo de pagamento para o cliente, fornecendo a possibilidade de pagamento da comanda através do aplicativo Android;
- Melhor controle para o estabelecimento e comodidade para o cliente ao utilizar mecanismos digitais para liberação dos clientes.

1.3 Estrutura do Trabalho

Para um melhor entendimento deste trabalho, realizamos a divisão em cinco capítulos, sendo eles:

No capítulo 1 é apresentado uma introdução sobre o projeto a ser desenvolvido durante esse trabalho.

No capítulo 2 são revisados os principais conceitos necessários para o entendimento deste trabalho.

No capítulo 3 são expostas as principais características do projeto, necessárias para a implementação do mesmo.

No Capítulo 4 são apresentados os resultados obtidos na implementação deste projeto.

No Capítulo 6 é apresentado a conclusão do projeto e as propostas de trabalhos futuros.

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

2.1 Sistemas Web

Com o intuito de fornecer um serviço que pode ser acessado de diversas localidades e a partir de qualquer dispositivo, uma solução web mostra se ideal. Utilizando um simples navegador web ou um aplicativo, pode se ter acesso a informações e realizar operações como consultas a banco de dados, acessar emails, compras, cadastros, registros de logs, etc. A Figura 2.1 ilustra a estrutura básica de um sistema web.

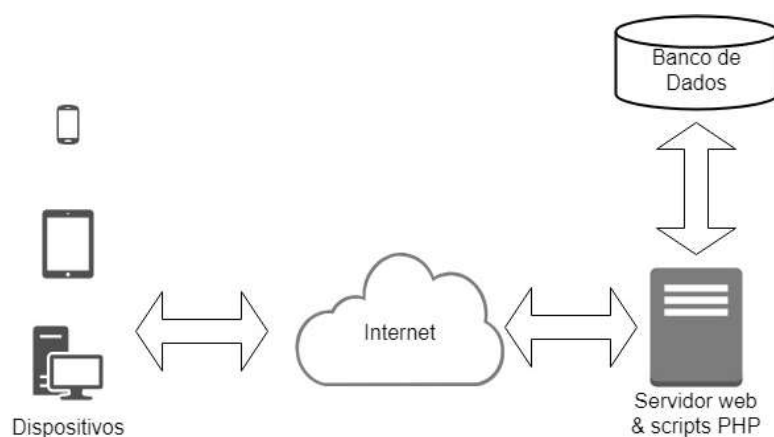


Figura 2.1: Estrutura de um sistema web

Um Sistema Web não se trata apenas de um site na internet, mas de uma solução que estabelece comunicação entre diversas aplicações diferentes (Banco de Dados, Servidor Web e Aplicativo) e entre outros sistemas, fazendo uso de uma linguagem de programação como por exemplo PHP, Python, Ruby entre outras. O Sistema Web também permite que novas aplicações interajam com outras já existentes e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis [4]. A troca de informação entre as aplicações deve ser feita através de mensagens padronizadas, XML ou JSON são os formatos mais utilizados.

Atualmente as soluções mais utilizadas para se construir um sistema web são SOAP

(*Simple Object Access Protocol*) ou REST (*Representational State Transfer*).

2.2 Servidor LAMP

Os Servidores web são responsáveis pela grande maioria dos serviços providos pela Internet, uma vez que hospedam todas as páginas e servem de base para todo tipo de sistemas web, incluindo webmails [5] e o escopo deste trabalho.

A pilha LAMP é um acrônimo para o sistema operacional Linux, o servidor web Apache, o banco de dados Mysql ou MariaDB e a linguagem de programação PHP. Tendo esses quatro componentes instalados é possível iniciar o desenvolvimento web. Vale lembrar que todos estes softwares citados são livres e de código aberto. Esta tecnologia também está disponível para o sistema operacional Windows com o acrônimo WAMP (*Windows Apache Mysql PHP*). Para ilustrar a estrutura da pilha LAMP, temos a Figura 2.2. Notamos que a pilha não fica restrita ao uso somente do php, outras linguagens também podem ser utilizadas, como pearl e python.

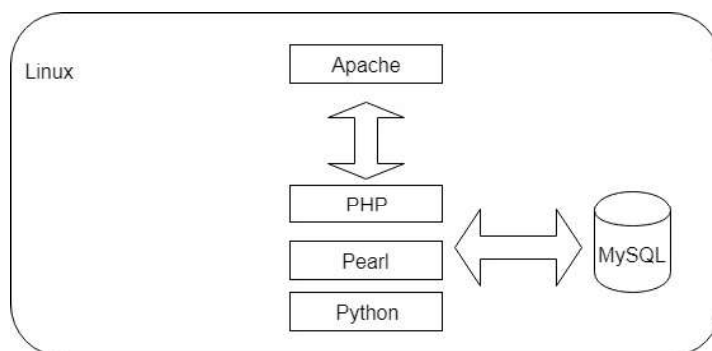


Figura 2.2: Componentes da pilha LAMP

2.2.1 Linux

Criado em 1991 por Linus Torvalds, o Linux é o kernel do sistema operacional GNU/Linux, derivado do Unix. O kernel é o núcleo de um sistema operacional, sendo responsável por fazer a comunicação entre hardware e software, gerenciando os recursos necessários para o bom funcionamento do software sem comprometer a integridade da máquina. Dentre os recursos gerenciados estão listados memória, processos, arquivos e todos os dispositivos periféricos [6]. Para ilustrar o papel do kernel, temos a Figura 2.3.

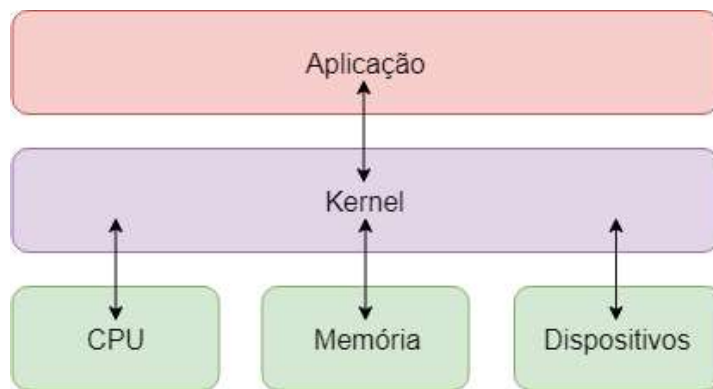


Figura 2.3: O kernel estabelece a comunicação entre aplicações e recursos de hardware

Diante dos grandes Softwares proprietários como DOS, MACOSX e o próprio Unix, o linux surgiu como software livre e de código aberto. Inspirados por projetos como GNU de Stallman, que defendia que softwares deveriam ser livres e de qualidade, estudantes de informática do mundo inteiro passaram a contribuir com o projeto [7].

Atualmente o linux possui diversas distribuições e é utilizado tanto em servidores quanto em desktops. O linux mantém sua ideologia de software livre e conta com uma grande comunidade disposta a trabalhar em melhorias, resolvendo bugs e adaptando o sistema para novos hardwares disponibilizados.

2.2.2 Apache

Da pilha LAMP, o apache que é o responsável por disponibilizar as páginas e seus conteúdos para os clientes. O apache é um servidor do protocolo HTTP (*Hypertext Transfer Protocol*), e é um dos mais bem sucedidos da história. Sua função primordial é oferecer suporte para páginas html e scripts CGI (*Computing Graphic Image*), as demais funcionalidades são expandidas através de módulos, passando a suportar scripts PHP, banco de dados, etc.

A Figura 2.4 mostra como uma página web é disponibilizada à uma máquina cliente através de uma requisição HTTP.

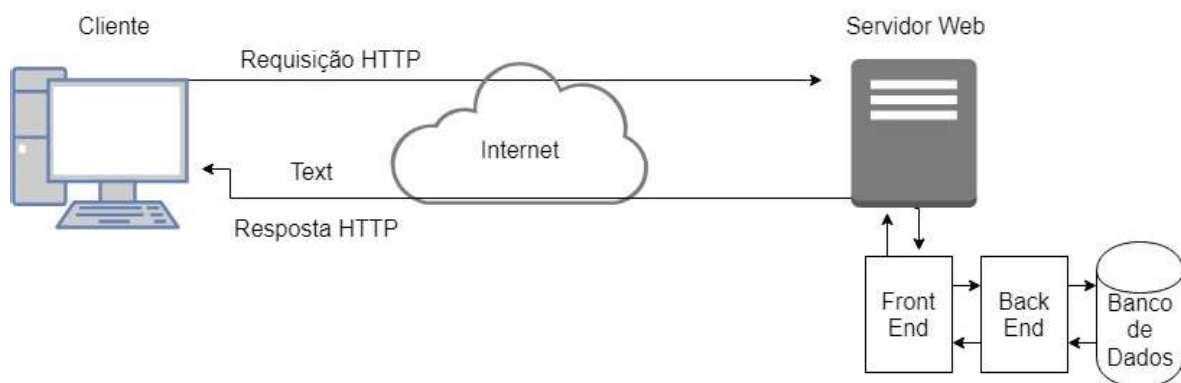


Figura 2.4: Funcionamento de um servidor web

Segundo uma pesquisa realizada pela Netcraft, o servidor apache foi a tecnologia dominante de 1996 até 2016, chegando ao ápice em novembro de 2005, com mais de 53 milhões de servidores, representando um total de 71%. Em uma pesquisa mais recente, em abril de 2018, foi revelado que mais de 450 milhões de servidores utilizam o apache, representando 25,58% dos servidores no mundo, ficando atrás apenas da microsoft com 36,94%[8].

2.2.3 MariaDB

Segundo Korth, banco de dados é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico [9].

Sendo assim, é no banco de dados (BD) , que ficam armazenados os dados do sistema, mas não simplesmente de forma aleatória e sim organizada pela relação entre esses dados. Em um banco de dados os dados são separados pelo programador em tabelas, podendo estar relacionadas com outras tabelas ou não.

MariaDB é um SGDB (*Sistema de Gerenciamento de Banco de Dados*) baseado no Mysql, inclusive mantendo os mesmos comandos, bibliotecas, interfaces e APIs [10], possibilitando uma fácil migração. MariaDB é de código aberto e colabora com inovadores como Alibaba, Google e Facebook para desenvolver e incorporar melhorias para toda a comunidade [11].

É através do SGDB que as aplicações tem acesso aos bancos e tabelas, podendo salvar novos dados, editar, remover, dar permissões, manter cópias, encriptar [12]. Ilustração presente na Figura 2.5.

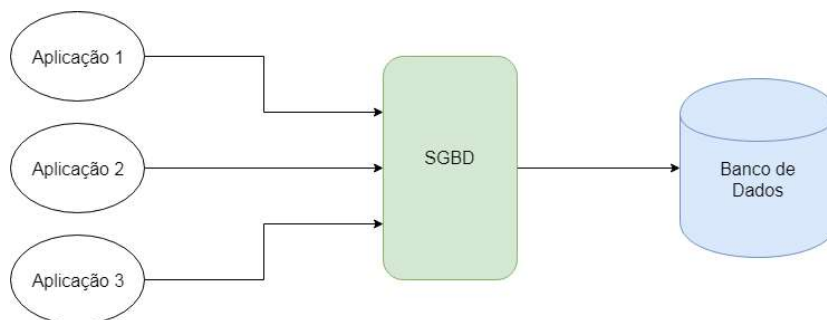


Figura 2.5: Estrutura de como uma aplicação acessa dados presentes no BD

Dentre as principais funções de um SGDB estão:

- Segurança

Restringir acesso às informações somente à usuários autorizados é de suma importância para a segurança de qualquer sistema, assim como a restrição de operações em níveis de acesso para cada tipo de usuário. Outro fator de segurança são as cópias para recuperação, imprescindíveis em caso de falhas ou até mesmo desastres que causem a perda dos dados.

- Controle de redundâncias

Um SGBD deve evitar que existam dados replicados em um sistema, os dados são armazenados em um só lugar impedindo que a duplicação desenfrede dados [13].

- Compartilhamento de dados

Para que não haja erros na leitura e escrita de dados o sgbd deve incluir um software que faça o controle de concorrência de dados. Assim mais de um usuário poderá fazer uso do sistema sem que possa prejudicar outro usuário.

- Interfaceamento

Um SGBD deve disponibilizar via interface os dados e tabelas, não obrigando o uso da aplicação que utiliza o BD para se ter acesso.

- Esquematização

Normalmente em bancos de dados tabelas podem estar relacionadas uma com as outras, portanto um SGBD deve ter mecanismos que compreendam esses relacionamentos [13].

- Integridade

A modificação de dados pela aplicação e usuários não autorizados devem ser negados. Dessa forma a segurança é reforçada e a integridade dos dados fica protegida.

- Backups

Bancos de Dados estão sujeitos a falhas de hardware, software, sabotagem e até mesmo desastres naturais, por isso deve fazer uso de cópias de segurança e através de mecanismos pré ajustados é possível recuperar os dados e evitar maiores danos.

2.2.4 PHP

A linguagem de programação tem papel fundamental para a criação do webservice, ela que fará o papel de intermediador entre a página web e outras aplicações e sistemas.

Atualmente php é um acrônimo recursivo para *PHP: hypertext processor*, é uma linguagem orientada a objetos muito utilizada para programação web, seu código pode ser facilmente embutido em um script html.

O código php é executado principalmente no servidor, assim pode interagir com outras aplicações rodando no servidor, como o banco de dados. Outra vantagem é que, sendo executado no servidor, o código fonte não fica exposto para todos que acessam uma página no domínio, senhas e outras informações confidenciais não serão expostas.

2.2.5 Scriptcase

O scriptcase é uma ferramenta do tipo RAD (*Rapid Application Development*) que visa acelerar o processo de desenvolvimento utilizando ambiente gráfico capaz de tirar o maior proveito do reuso de código. Através da sua interface é possível gerar códigos PHP de estruturas já criadas ou salvas. Dessa forma, é possível concentrar em partes mais importantes do desenvolvimento.

2.2.6 phpMyAdmin

PhpMyAdmin é uma ferramenta que permite a criação, leitura, atualização e deleção (CRUD) de tabelas e bancos de dados, tudo por meio de uma interface gráfica. Com o intuito de agilizar esses processos, phpMyAdmin é uma ferramenta poderosa, mostrando excelente praticidade.

O phpMyAdmin tem seu código escrito em php e é acessado pelo navegador web. Foi construído para lidar com a administração do SGBD MySQL, suportando também uma ampla gama de operações no MariaDB.

O projeto phpMyAdmin é membro da SFC (*Software Freedom Conservancy*) que é uma organização sem fins lucrativos que ajuda a promover, desenvolver e defender softwares gratuitos livres e código aberto (FLOSS) [14].

2.3 Linguagens Utilizadas

Várias linguagens são utilizadas no desenvolvimento web, veremos nas seções abaixo o papel de cada uma delas, em resumo, o HTML cuidará do conteúdo das páginas, seguido do CSS, responsável pelo layout, isto é, posicionamento, cores e tamanhos de cada elemento, o Javascript dará vida às páginas cuidando do comportamento de seus elementos (Figura 2.6). Essas três primeiras linguagens são destinadas à parte front-end da aplicação e são executadas no lado cliente da aplicação, pelo navegador web poupando processamento por parte do servidor. Por fim o PHP fará as operações de back-end.

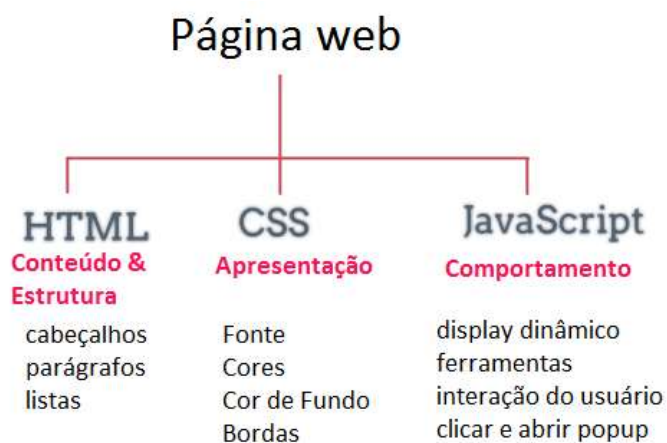


Figura 2.6: Funções de cada linguagem, HTTP, CSS e Javascript

Para o desenvolvimento mobile utilizaremos o react-native, uma linguagem baseada em javascript que possibilita a criação de aplicativos Android e IOS com o mesmo código.

2.3.1 HTML

O HTML é uma linguagem de marcação de hiper texto, um código em HTML é inteiramente estruturado dentro de tags, essas tags indicam se o texto entre elas será do tipo título, cabeçalho, parágrafo, hyperlink, botão e diversas outras. Tags também podem indicar seções, áreas e subdivisões dentro da página.

O uso do HTML é direcionado mais especificamente ao conteúdo que será apresentado dentro de uma página, ações e layout são deixados para outras linguagens.

2.3.2 CSS

Desde a criação do HTML não era intencional que houvessem tags para formatação de páginas. Foram adicionadas fontes e cores ao HTML na sua versão 3.2, porém essas tags deveriam ser adicionadas em cada página de um site, tornando o processo mais demorado e custoso.

A partir dessa necessidade foi criado o CSS (*Cascading Style Sheets*), onde era possível mudar o visual de um site inteiro alterando apenas um arquivo externo .css, aumentando a produtividade do programador. Dentre as funcionalidades do CSS estão incluídas mudanças de fontes, tamanhos, cores, posicionamento de elementos e outras funções relacionadas à aparência, deixando para que o HTML se tornasse responsável pelo conteúdo da página.

2.3.3 JavaScript

JavaScript uma linguagem de programação orientada a objetos, é responsável pelas ações que ocorrem em uma página web, podem ser alterados estilos e imagens assim como mostrar submenus ou mensagens em novas janelas, tudo isso com simples interações do usuário tais como um clique ou o passar do ponteiro do mouse sobre determinada seção. O mais importante é isso pode ser feito sem que seja necessário recarregar outra página inteira, deixando o acesso mais rápido e leve[15].

O uso do JavaScript não é restrito apenas às páginas web, alguns SGBD, MongoDB e CouchDB utilizam o JavaScript em seus códigos fonte[16]. Ao contrário do que se imagina, a linguagem JavaScript não está relacionada com a linguagem Java.

2.3.4 React Native

React Native é uma biblioteca em JavaScript, desenvolvida pelo Facebook com o intuito de reduzir tempo e custo na fabricação de aplicativos nativos para Android e iOS. Com React Native é possível criar aplicações multiplataforma para Android e iOS com um único código.

Um aplicativo nativo tem melhor performance em comparação com aplicativos híbridos e web, podem fazer uso de recursos do aparelho como câmera, GPS, acelerômetro e giroscópio com maior desempenho. Ilustração na Figura 2.7.

Grandes aplicações como Facebook, Instagram, Baidu e Airbnb já utilizam o React Native para desenvolver suas aplicações.

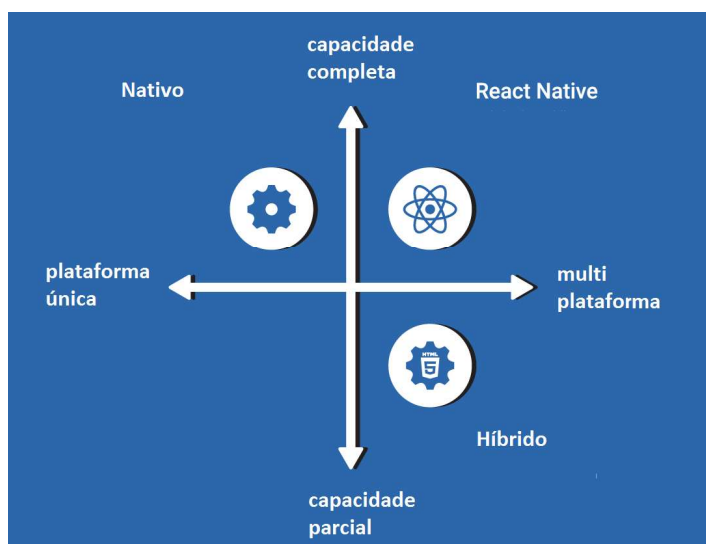


Figura 2.7: Vantagens de se utilizar o react-native

2.4 Android

O Android é um sistema operacional de código aberto e baseado em Linux para dispositivos móveis, como smartphones e tablets. O Android foi desenvolvido pela *Open Handset Alliance*, liderada pelo Google e por outras empresas. O Android oferece uma abordagem unificada para o desenvolvimento de aplicativos para dispositivos móveis, o que significa que os desenvolvedores precisam desenvolver apenas para o Android, e seus aplicativos devem poder ser executados em diferentes dispositivos equipados com esse sistema operacional. A primeira versão beta do SDK (*Software Development Kit*) foi lançada pelo Google em 2007, enquanto a primeira versão comercial, o Android 1.0, foi lançada em setembro de 2008.

Atualmente, o sistema operacional do Android é o mais utilizado no mundo [17]. O Android é popular porque suporta grande número de aplicativos. Como era de código aberto e muito favorável aos desenvolvedores, os desenvolvedores o adotaram. Isso permite que eles desenvolvam seus aplicativos no Android SDK. Para os usuários, o Android é fácil de usar e bastante configurável.

2.4.1 Android Studio

Como descrito em [17], O Android Studio é o ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de aplicativos Android e é baseado no *IntelliJ IDEA*. Além do editor de código e das ferramentas de desenvolvedor avançados do *IntelliJ IDEA*, o Android Studio oferece ainda mais recursos para aumentar sua produtividade na criação de aplicativos Android, como:

- Um sistema de compilação flexível baseado no Gradle;
- Um emulador rápido com inúmeros recursos;
- Um ambiente unificado para você poder desenvolver para todos os dispositivos Android;
- Instant Run para aplicar alterações a aplicativos em execução sem precisar compilar um novo APK;
- Modelos de códigos e integração com GitHub para ajudar a criar recursos comuns dos aplicativos e importar exemplos de código;
- Ferramentas e estruturas de teste cheias de possibilidades;
- Ferramentas de verificação de código suspeito para detectar problemas de desempenho, usabilidade, compatibilidade com versões e outros;
- Compatibilidade com C++ e NDK;

- Compatibilidade embutida com o Google Cloud Platform, facilitando a integração do Google Cloud Messaging e do App Engine;

2.5 UML

O *Unified Modeling Language* (UML) foi desenvolvida por Grady Booch, Jim Rumbaugh, and Ivar Jacobson nos anos 90. Segundo [Booch, 2005], a UML é uma linguagem padrão para elaboração da documentação padronizada de um software. Durante sua criação, foram mescladas diversas notações concorrentes que eram usadas pela indústria de desenvolvimento de software. A UML é usada para visualização, especificação, documentação e construção de artefatos para sistemas complexos de software.

Desde sua criação a UML tem evoluído e gerado diversas versões, atualmente entra-se na versão 2.5.1. Vale destacar que desde sua versão 2.3 a UML se tornou um padrão ISO. Em sua versão 2.5, os diagramas UML são divididos em 2 tipos principais de diagramas: diagramas estruturais e diagramas comportamentais.

- Diagramas Estruturais

Nestes diagramas são enfatizados os aspectos estruturais do sistema a ser modelado. Por representarem a estrutura, esses diagramas são usados para a visualizar, especificar e documentar aspectos estáticos do sistema. Podem ser representados pelos seguintes diagramas:

1. Diagrama de Classes;
2. Diagramas de Objetos;
3. Diagramas de Componentes;
4. Diagramas de Implantação;
5. Diagramas de Pacotes;
6. Diagramas de Estrutura;
7. Diagramas de Perfil;

- Diagramas Comportamentais

Nestes diagramas são enfatizados o que deve acontecer no sistema a ser modelado. Por representarem o comportamento, esses diagramas são usados para descrever a funcionalidade do software. Os diagramas comportamentais contém um subconjunto de diagramas chamado de diagramas de interação. Nestes diagramas, o foco é no controle do fluxo de controle e dados. Um exemplo de diagrama de interação é o diagrama de sequência.

1. Diagrama de Caso de Uso;

2. Diagrama de Sequência;
3. Diagrama de Comunicação;
4. Diagrama de Estados;
5. Diagrama de Atividade;
6. Diagrama de Tempo;
7. Diagramas de Visão Geral de Interação;

Para melhor entendimento do sistema proposto neste projeto, realizamos a elaboração do diagrama de caso de uso.

2.5.1 Diagrama de Caso de Uso

O diagrama de caso de uso é a representação das interações entre os agentes e diferentes casos de uso do sistema. Através deste diagrama é possível descrever as funcionalidades do sistema do ponto de vista de cada ator. O diagrama de caso de uso é representado por:

- **Atores**

O ator é um usuário do sistema, pode ser uma pessoa, grupo, cargo, organização ou outro sistema externo que desempenha um papel em uma ou mais interações com o sistema. Geralmente é representado por um boneco contendo um identificador.

- **Caso de uso**

Representa uma função ou ação do sistema que fornece um valor mensurável para os atores. Sua representação é através de uma elipse contendo um rótulo com o nome do caso de uso.

- **Relacionamento entre os elementos**

Os relacionamentos entre os elementos ajudam a descrever a relação entre os atores e os casos de uso. O relacionamento entre os elementos do diagrama de caso de uso podem ser:

1. **Associação entre atores e casos de uso**

Demonstra a funcionalidade do sistema do ponto de vista do ator (usuário).

2. **Generalizações entre atores;**

Representa a herança entre os atores. Quando existir um relacionamento entre os atores, os casos de uso relacionados com o ator filho também serão casos de uso do ator pai.

3. Generalizações, extends e includes entre os casos de uso;

Estas associações são utilizadas para explicitar relação entre os casos de uso. Os includes indicam que o caso de uso é essencial para o comportamento de outro caso de uso. Os extends são o oposto dos includes, ou seja, representam que a relação entre os casos de uso não são essenciais. Desta forma, os casos de uso que possuem o extend apenas acrescentam funcionalidades a outros casos de uso.

2.6 Modelo Entidade-Relacionamento

O *modelo entidade-relacionamento* (MER) foi definido por Peter Chen em 1976 em seu paper [18]. Segundo Chen, a visão de uma dada realidade, baseia-se no relacionamento entre conceitos desta realidade, os quais retratam os fatos que governam esta mesma realidade, e que cada conceito (entidade ou relacionamento) pode possuir atributos (qualificadores desta realidade). O MER consiste em um modelo para descrever os dados, regras de negócio ou requisitos do processo.

O MER, normalmente, é o resultado de uma análise sistemática para definir e descrever o que é importante para os processos. Geralmente é utilizado para representar a forma abstrata que o banco de dados irá possuir. Sua representação é feita através de caixas (entidades) que são conectadas através de associações e dependências. Sua composição é dada por três elementos: entidades, relacionamentos e atributos.

- Entidades

São objetos ou partes envolvidas. As entidades podem ser objetos físicos (como um carro ou prédio), um evento (como a venda de um carro) ou um conceito (como um pedido em um restaurante).

- Relacionamentos

Relaciona como as entidades interagem umas às outras. Podem ser pensados como verbos ligando dois ou mais substantivos. Existem em três formas: relacionamento um para um (1..1), um para muitos (1..n ou 1..*) ou muitos para muitos (n..n ou *..*).

- Atributos

São características ou qualidades de uma entidade que possuam valor para o negócio. Por exemplo, um cliente possui nome, endereço e telefone. Os atributos são identificados durante a análise de requisitos e podem ser classificados como descritivos ou nominativos.

2.6.1 Dicionário de Dados

Juntamente ao MER, é necessário que seja criado um documento com a explicação de todos os objetos criados nesse modelo. Esse documento é chamado de dicionário de dados e serve como referência textual sobre todos os objetos mostrados no MER.

No dicionário de dados cada objeto é descrito de forma mais detalhada e, com isso, permite o melhor entendimento do modelo proposto. Em nosso projeto, iremos criar o dicionário de dados contendo as informações do nome da entidade, tipo e tamanho, se o campo é obrigatório e uma descrição sobre a entidade.

Capítulo 3

METODOLOGIA E ARTEFATOS

Neste capítulo aborda a metodologia e a criação dos artefatos utilizada no trabalho, detalhando as características do projeto como delimitação do tema, estudo sobre o funcionamento de casas noturnas, elaboração da proposta e a criação dos artefatos. Devido a explicação detalhada de cada artefato, este capítulo se torna mais extenso que os demais.

3.1 Delimitação do Tema

Este projeto propõem o desenvolvimento de um sistema Web para os estabelecimentos comerciais como casas noturnas, bares e restaurantes e um aplicativo Android para que os clientes possam realizar o pagamento de suas comandas sem ter a necessidade de aguardar o atendente ou esperar em uma fila.

O sistema Web foi necessário devido ao custo e a dificuldade de integração do aplicativo com os sistemas de empresas já presentes no mercado. Dessa forma, a solução proposta está inicialmente delimitada à solução fechada deste trabalho, ou seja, sua integração está apenas entre o sistema web e o aplicativo Android proposto.

O aplicativo Android fornecerá a funcionalidade de cadastramento de usuário, pagamento e validador de saída. Optou-se por utilizar um cenário de um estabelecimento de casa noturna pois envolve uma maior gama de ações para o aplicativo. Após a consolidação nesse ambiente proposto, a expansão do aplicativo para bares e restaurantes se torna mais fácil.

O desenvolvimento do tema proposto foi dividido em 7 etapas, sendo elas:

- Etapa 1 - Estudo sobre o funcionamento de casas noturnas:

Nesta etapa foi realizada a avaliação do funcionamento e as funcionalidades necessárias dos estabelecimentos de casas noturnas para o sistema web.

- Etapa 2 - Elaboração de proposta:

Nesta etapa foi feito um estudo sobre as vantagens do uso do aplicativo e de como seria utilizado.

- Etapa 3 - Preparação do ambiente:

Instalação e configuração do servidor LAMP.

- Etapa 4 - Criação dos artefatos:

Elaboração do diagrama de caso de uso e modelo entidade-relacionamento para padronizar e facilitar o desenvolvimento do sistema Web e do aplicativo Android.

- Etapa 5 - Criação do Banco de Dados conforme MER proposto:

Criação do banco de dados e suas tabelas conforme proposto pelo MER.

- Etapa 6 - Desenvolvimento do sistema Web:

Desenvolvimento da aplicação Web utilizando a plataforma de desenvolvimento script-case conforme requisitos levantados e criação dos diagramas de casos de uso.

- Etapa 7 - Desenvolvimento do aplicativo Android:

Implementação dos três módulos do aplicativo Android integrado ao sistema Web criado. Os três módulos são: cadastramento de usuário, pagamento de comanda e validação de saída através de código de comanda.

3.2 Estudo sobre o funcionamento de casas noturnas

Em geral, ao chegar em casas noturnas os clientes são recebidos em uma recepção onde é feito o seu cadastro e é entregue uma comanda para marcação do consumo dentro do estabelecimento.

No ambiente existe uma área para o bar, onde são feitos os pedidos, marcados na comanda e em seguida entregue os consumíveis.

Ao final da noite o cliente deve se dirigir ao caixa para realizar o pagamento, nesse processo ocorre a soma de valores, conferência de itens e o pagamento, logo em seguida a saída é liberada para o cliente. Percebe-se então dois pontos principais onde o tempo pode ser otimizado, na recepção e no caixa final para o acerto da conta.

Tendo em vista essa rotina, as funcionalidades que devem estar presentes no sistema web são:

- Cadastro de clientes e geração de comandas
- Cadastro de produtos

- Editar itens das comandas do sistema, incluindo adicionar, alterar, visualizar e remover.
- Soma de itens e fechamento de conta.
- Sistema de pagamento.

3.3 Elaboração de Proposta

Com a finalidade de prover todos os recursos existentes em modelos tradicionais e criar uma base de apoio para o aplicativo proposto, foi desenvolvido um sistema web que possui as funcionalidades comuns de qualquer sistema que gerencia estabelecimentos de casa noturnas. Dessa forma, o modelo tradicional de pedido, controle de comanda, pagamento e cadastro é atendida. Além de, como dito anteriormente, servir de apoio caso ocorram falhas decorrentes do aplicativo ou de smartphones de clientes.

Através do aplicativo, o próprio cliente pode fazer seu cadastro e visualizar sua comanda virtual em seu aparelho celular. Um código é gerado para esse cliente e em seguida deve ser apresentado para que sua entrada no estabelecimento seja liberada. O cliente torna-se mais satisfeito ao gastar menos tempo na fila de entrada, assim pode aproveitar mais do estabelecimento.

Para marcação de consumo é apresentado o código ao atendente do bar, no balcão. Os itens consumidos são registrados no sistema e automaticamente na comanda virtual presente no aplicativo. Em caso de extravio do aparelho ou esgotamento de bateria, o cliente deve se dirigir ao caixa para seu código e receber uma comanda manual com os itens consumidos até o momento marcados.

Ao final o pagamento pode ser efetuado via aplicativo, a soma de valores e a conferência de itens estará sendo feita a todo momento a partir da entrada. O cliente terá controle sobre os valores que estão sendo gastos. Sua saída do estabelecimento será liberada se em seu código estiver constando que o pagamento foi efetuado. O cliente sai satisfeito, por não demorar na fila e o estabelecimento ganhará mais prestígio por proporcionar controle e agilidade.

3.4 Criação dos artefatos

Para auxiliar o desenvolvimento do sistema web e do aplicativo foram criados os artefatos, diagrama de caso de uso, especificações de caso de uso e modelo entidade-relacionamento. Todos os artefatos criados estão listados nessa seção.

3.4.1 Diagrama de Caso de Uso

O diagrama de caso de uso mapeia os atores e suas respectivas ações. Tendo a esquematização da Figura 3.1 em mãos, nenhuma funcionalidade projetada será esquecida pelo programador.

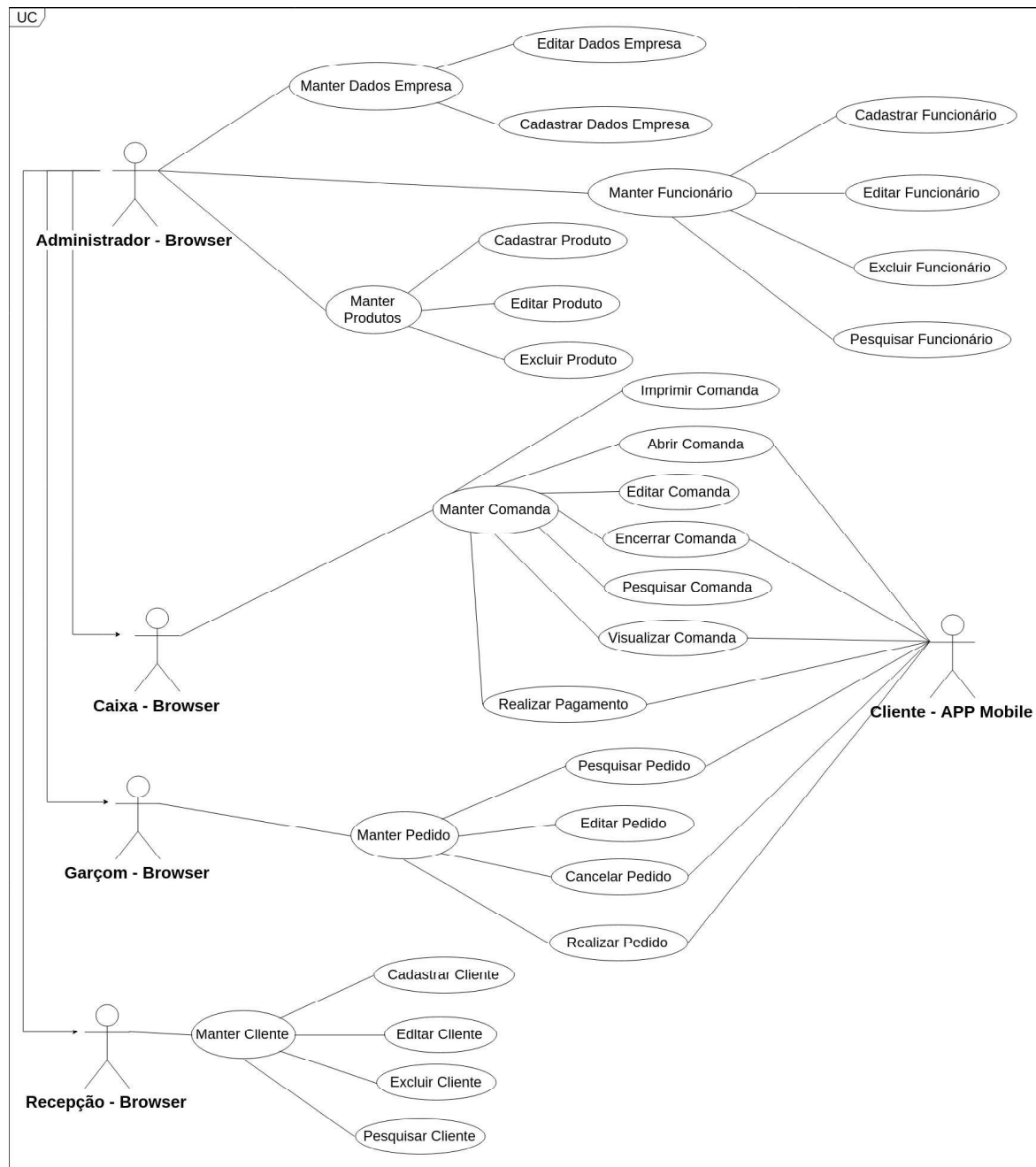


Figura 3.1: Diagrama de Caso de Uso

Tabela 3.1: Atores dos casos de uso

PAPEL	RESPONSABILIDADE
Administrador	Responsável por administrar o sistema. Realiza o cadastro das informações da empresa, como por exemplo, conta bancária, CNPJ, produtos, preços, entre outros. É responsável pela criação dos garçons. Por padrão, possui todas as permissões dos usuários Caixa, Balcão e Recepção.
Caixa	Usuário responsável por realizar o fechamento da comanda do cliente. Realiza a impressão da comanda para que o cliente possa revisar os gastos. Caso necessário, pode cancelar ou editar pedidos errados. Por fim, recebe o pagamento do cliente.
Garçom	Usuário responsável por atender os pedidos dos clientes. Ao receber um pedido o usuário com perfil de garçom cadastra o pedido na comanda do cliente. Finalizado o cadastramento do pedido, o atendente irá entregar o produto solicitado ao cliente.
Recepção	Usuário responsável por realizar o cadastro inicial do cliente no sistema do estabelecimento, caso o cliente não o tenha feito através do aplicativo para Smartphone.
Cliente	Responsável por realizar seu cadastro, seja através do aplicativo para Smartphone ou através da recepção, e abertura de comanda. Realiza os pedidos diretamente no balcão do estabelecimento, pode realizar o fechamento da comanda e, caso queira, poderá realizar o pagamento através do aplicativo utilizando cartão de crédito.

Em seguida temos as especificações de casos de uso, nelas cada detalhe de cada funcionalidade é definida, mapeando o fluxo do trabalho do programador.

3.4.1.1 Especificação dos casos de uso

Tabela 3.2: Caso de Uso - Cadastrar Cliente

Nome do Caso de Uso	Cadastrar Cliente
Autor(es) envolvidos	Administrador / Recepção
Resumo	Ação de cadastrar um novo cliente ao sistema
Pré-Condições	Estar autenticado como "administrador" ou usuário "Recepção"
Pós-Condições	Um novo usuário com perfil de cliente será adicionado a base de dados
Fluxo Básico	<ol style="list-style-type: none">1. Clicar menu "Segurança".2. Selecionar a opção "Usuários".3. Clicar no botão "Novo".4. Preencher os campos solicitados e clicar em "Salvar".5. O sistema irá validar as informações inseridas.6. Uma mensagem de cadastro realizado com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none">1. O ator clica em "Sair".2. O sistema encerra a ação de cadastrar cliente.3. O sistema retorna a tela anterior.
Fluxo de Exceções	<ol style="list-style-type: none">1. O ator não preenche os campos com valores válidos.2. O sistema emite uma mensagem de erro.3. O sistema retorna a anterior ao erro.

Tabela 3.3: Caso de Uso - Pesquisar Cliente

Nome do Caso de Uso	Pesquisar Cliente
Autor(es) envolvidos	Administrador / Recepção
Resumo	Pesquisa um cliente no sistema
Pré-Condições	Estar autenticado como "administrador" ou usuário "Recepção"
Pós-Condições	Exibe dados do cliente pesquisado
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Segurança". 2. Selecionar a opção "Usuários". 3. O ator realiza a busca do usuário desejado no campo "Busca rápida". 4. O sistema exibe o ícone de um lápis para "Editar Cliente".
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica no símbolo de lápis para "Editar Cliente". 2. O sistema abre o caso de uso para editar o cliente.
Fluxo Alternativos A2	<ol style="list-style-type: none"> 1. O ator clica em "Novo". 2. O sistema abre o caso de uso para "Cadastrar Cliente".
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.4: Caso de Uso - Editar Cliente

Nome do Caso de Uso	Editar Cliente
Autor(es) envolvidos	Administrador / Recepção
Resumo	Ação de editar os dados de um cliente já cadastrado no sistema
Pré-Condições	Estar autenticado como "administrador" ou usuário "Recepção" e o cliente a ser editado já ter sido criado
Pós-Condições	Os dados do cliente serão alterados para o novo valor inserido
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Segurança". 2. Selecionar a opção "Usuários". 3. Selecionar o ícone de lápis do cliente desejado. 4. Alterar o(s) campo(s) desejados e clicar em "Salvar". 5. O sistema irá validar as informações inseridas. 6. Uma mensagem de alteração realizada com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Sair". 2. O sistema encerra a ação de editar cliente. 3. O sistema retorna a tela anterior.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.5: Caso de Uso - Excluir Cliente

Nome do Caso de Uso	Excluir Cliente
Autor(es) envolvidos	Administrador / Recepção
Resumo	Ação de remover um cliente do sistema
Pré-Condições	Estar autenticado como "administrador" ou usuário "Recepção"
Pós-Condições	O usuário selecionado será removido da base de dados
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Segurança". 2. Selecionar a opção "Usuários". 3. Selecionar o ícone de lápis do cliente desejado. 4. Selecionar a opção de "Excluir". 5. Uma mensagem de cliente excluído com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Sair". 2. O sistema encerra a ação de excluir cliente. 3. O sistema retorna a tela anterior.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O sistema não consegue excluir o cliente solicitado. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.6: Caso de Uso - Cadastrar Produto

Nome do Caso de Uso	Cadastrar Produto
Autor(es) envolvidos	Administrador
Resumo	Ação de cadastrar um novo produto ao estabelecimento
Pré-Condições	Estar autenticado como "administrador"
Pós-Condições	O produto será adicionado a lista de produtos ofertados
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Cardápio". 2. Clicar no botão "Cadastrar Produto". 3. Preencher os campos solicitados e clicar em incluir. 4. Uma mensagem de cadastro realizado com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em cancelar. 2. O sistema encerra a ação de cadastrar um novo produto. 3. O sistema retorna as opções anteriores ao cadastro de produto.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.7: Caso de Uso - Editar Produto

Nome do Caso de Uso	Editar Produto
Autor(es) envolvidos	Administrador
Resumo	Ação de editar os dados de um produto já cadastrado no sistema
Pré-Condições	Estar autenticado como "administrador" e o produto a ser editado já ter sido criado
Pós-Condições	Os dados do produto serão alterados para o novo valor inserido
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Cardápio". 2. Selecionar o produto desejado. 3. Clicar no ícone de lápis. 4. Alterar o(s) campo(s) desejados e clicar em "Salvar". 5. O sistema irá validar as informações inseridas. 6. Uma mensagem de alteração realizada com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em cancelar. 2. O sistema encerra a ação de editar produto. 3. O sistema retorna a tela principal do menu produto.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos.. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.8: Caso de Uso - Excluir Produto

Nome do Caso de Uso	Excluir Produto
Autor(es) envolvidos	Administrador
Resumo	Ação de remover um produto do sistema
Pré-Condições	Estar autenticado como "administrador"
Pós-Condições	O produto selecionado será removido da base de dados
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Cardápio". 2. Selecionar o produto desejado. 3. Clicar no ícone de lápis. 4. Selecionar a opção de "Excluir". 5. Uma mensagem de cliente excluído com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Sair". 2. O sistema encerra a ação de excluir cliente. 3. O sistema retorna a tela anterior.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O sistema não consegue excluir o produto solicitado. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.9: Caso de Uso - Cadastrar Dados Empresa

Nome do Caso de Uso	Cadastrar Dados Empresa
Autor(es) envolvidos	Administrador
Resumo	Ação de cadastrar os dados do estabelecimento
Pré-Condições	Estar autenticado como "administrador"
Pós-Condições	Os dados de conta, CNPJ, razão social, endereço, telefones, serão adicionados ao perfil do estabelecimento
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar no menu "Empresa". 2. Preencher os campos solicitados. 3. Clicar em salvar 4. Uma mensagem de cadastro realizado com sucesso será exibida.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.10: Caso de Uso - Editar Dados Empresa

Nome do Caso de Uso	Editar Dados Empresa
Autor(es) envolvidos	Administrador
Resumo	Ação de editar os dados do estabelecimento
Pré-Condições	Estar autenticado como "administrador"
Pós-Condições	Os dados do estabelecimento serão alterados para o novo valor inserido
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar no menu "Empresa". 2. Alterar os campos desejados. 3. Clicar em salvar. 4. Uma mensagem de cadastro realizado com sucesso será exibida.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.11: Caso de Uso - Cadastrar Funcionário

Nome do Caso de Uso	Cadastrar Funcionário
Autor(es) envolvidos	Administrador
Resumo	Ação de cadastrar um novo funcionário ao sistema
Pré-Condições	Estar autenticado como "administrador"
Pós-Condições	Um novo usuário com o perfil desejado (garçom, balcão, caixa ou recepção) será adicionado a base de dados
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Segurança". 2. Selecionar a opção "Usuários". 3. Clicar no botão "Novo". 4. Preencher os campos solicitados e clicar em "Salvar". 5. O sistema irá validar as informações inseridas. 6. Uma mensagem de cadastro realizado com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Sair". 2. O sistema encerra a ação de cadastrar funcionário. 3. O sistema retorna a tela anterior.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.12: Caso de Uso - Pesquisar Funcionário

Nome do Caso de Uso	Pesquisar Funcionário
Autor(es) envolvidos	Administrador
Resumo	Pesquisa um funcionário no sistema
Pré-Condições	Estar autenticado como "administrador"
Pós-Condições	Exibe dados do funcionário pesquisado
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Segurança". 2. Selecionar a opção "Usuários". 3. O ator realiza a busca do usuário desejado no campo "Busca rápida". 4. O sistema exibe o ícone de um lápis para editar o funcionário.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica no símbolo de lápis para "Editar Funcionário". 2. O sistema abre o caso de uso para editar o funcionário.
Fluxo Alternativos A2	<ol style="list-style-type: none"> 1. O ator clica em "Novo". 2. O sistema abre o caso de uso para "Cadastrar Funcionário".
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.13: Caso de Uso - Editar Funcionário

Nome do Caso de Uso	Editar Funcionário
Autor(es) envolvidos	Administrador
Resumo	Ação de editar os dados de um funcionário já cadastrado no sistema
Pré-Condições	Estar autenticado como "administrador" e o funcionário a ser editado já ter sido criado
Pós-Condições	Os dados do funcionário serão alterados para o novo valor inserido
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Segurança". 2. Selecionar a opção "Usuários". 3. Selecionar o ícone de lápis do funcionário desejado. 4. Alterar o(s) campo(s) desejados e clicar em "Salvar". 5. O sistema irá validar as informações inseridas. 6. Uma mensagem de alteração realizada com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Sair". 2. O sistema encerra a ação de editar funcionário. 3. O sistema retorna a tela principal do menu funcionário.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.14: Caso de Uso - Excluir Funcionário

Nome do Caso de Uso	Excluir Funcionário
Autor(es) envolvidos	Administrador
Resumo	Ação de remover um funcionário do sistema
Pré-Condições	Estar autenticado como "administrador"
Pós-Condições	O funcionário selecionado será removido da base de dados
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Segurança". 2. Selecionar a opção "Usuários". 3. Selecionar o ícone de lápis do funcionário desejado. 4. Selecionar a opção de "Excluir". 5. Uma mensagem de funcionário excluído com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em cancelar. 2. O sistema encerra a ação de excluir funcionário. 3. O sistema retorna a tela da opção "Pesquisar Funcionário".
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O sistema não consegue excluir o funcionário solicitado. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.15: Caso de Uso - Realizar Pedido

Nome do Caso de Uso	Realizar Pedido
Autor(es) envolvidos	Balcão / Cliente
Resumo	Ação de realizar um novo pedido
Pré-Condições	Estar autenticado como "cliente" ou usuário de perfil "balcão"
Pós-Condições	O pedido será adicionado a lista de pedidos realizados
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Comanda". 2. Selecionar a opção "Realizar Pedido". 3. O ator seleciona o produto desejado através do seu código. 4. O ator seleciona o código da comanda do cliente. 5. Uma mensagem de pedido realizado com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em cancelar. 2. O sistema encerra a ação de realizar um novo pedido. 3. O sistema retorna a tela inicial do menu "Comanda".
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.16: Caso de Uso - Editar Pedido

Nome do Caso de Uso	Editar Pedido
Autor(es) envolvidos	Administrador / Balcão / Cliente
Resumo	Ação de alterar um pedido realizado no sistema.
Pré-Condições	Estar autenticado como "administrador", usuário com perfil "Balcão" ou cliente e o pedido a ser alterado não ter entrado no status de "Em Preparo"
Pós-Condições	Os dados do pedido serão alterados para o novo valor inserido
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Comanda". 2. Selecionar a opção "Editar Pedido". 3. Alterar o(s) campo(s) desejados e clicar em "Salvar". 4. O sistema irá validar as informações inseridas. 5. Uma mensagem de alteração realizada com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Sair". 2. O sistema encerra a ação de editar pedido. 3. O sistema retorna a tela principal do menu comanda.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.17: Caso de Uso - Excluir Pedido

Nome do Caso de Uso	Excluir Pedido
Autor(es) envolvidos	Administrador / Balcão / Cliente
Resumo	Ação de cancelar um pedido realizar
Pré-Condições	Estar autenticado como "administrador", usuário com perfil "Balcão" ou cliente e o pedido a ser alterado não ter entrado no status de "Em Preparo"
Pós-Condições	O pedido selecionado será cancelado
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Comanda". 2. Selecionar a opção "Editar Pedido". 3. O ator seleciona os valores necessários. 4. Selecionar a opção de "Excluir Pedido". 5. Uma mensagem de pedido excluído com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Sair". 2. O sistema encerra a ação de excluir pedido. 3. O sistema retorna a tela principal do menu comanda.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O sistema não consegue excluir o pedido solicitado. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.18: Caso de Uso - Pesquisar Pedido

Nome do Caso de Uso	Pesquisar Pedido
Autor(es) envolvidos	Administrador / Balcão / Cliente
Resumo	Pesquisa um garçom no sistema
Pré-Condições	Estar autenticado como "administrador", usuário com perfil "Balcão" ou Cliente
Pós-Condições	Exibe dados do pedido pesquisado
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Comanda". 2. O ator realiza a busca através da busca rápida ou filtro dinâmico utilizando o código da comanda do cliente.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Editar Pedido". 2. O sistema abre o caso de uso para editar o pedido.
Fluxo Alternativos A2	<ol style="list-style-type: none"> 1. O ator clica em "Realizar Pedido". 2. O sistema abre o caso de uso para realizar pedido.
Fluxo Alternativos A3	<ol style="list-style-type: none"> 1. O ator clica em "Abrir Comanda". 2. O sistema abre o caso de uso para abrir comanda.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.19: Caso de Uso - Abrir Comanda

Nome do Caso de Uso	Abrir Comanda
Autor(es) envolvidos	Administrador / Caixa / Cliente
Resumo	Ação de realizar abrir uma nova comanda
Pré-Condições	Estar autenticado como "administrador", Cliente ou usuário de perfil "balcão". No caso de Cliente, não poderá ter outra comanda em aberto, ou seja, só poderá abrir uma nova comanda caso todas as anteriores estejam fechadas e pagas.
Pós-Condições	Uma comanda individual será aberta para o cliente que entrar no estabelecimento.
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar no menu "Comanda". 2. Selecionar a opção "Abrir Comanda". 3. Uma nova comanda individual será aberta e vinculada ao cliente. 4. Uma mensagem de comanda aberta com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Sair". 2. O sistema encerra a ação de abrir uma nova comanda. 3. O sistema retorna a tela inicial sistema.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.20: Caso de Uso - Editar Comanda

Nome do Caso de Uso	Editar Comanda
Autor(es) envolvidos	Administrador / Balcão
Resumo	Ação de alterar a comanda de um cliente no sistema.
Pré-Condições	Estar autenticado como "administrador" ou usuário com perfil "Balcão"
Pós-Condições	Os dados da comanda serão alterados para o(s) novo(s) valor(es) inserido(s)
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Comanda". 2. Selecionar a opção "Editar Pedido". 3. Alterar o(s) campo(s) desejados e clicar em "Salvar". 4. O sistema irá validar as informações inseridas. 5. Uma mensagem de alteração realizada com sucesso será exibida.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Sair". 2. O sistema encerra a ação de editar comanda. 3. O sistema retorna a tela principal do menu comanda.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.21: Caso de Uso - Encerrar Comanda

Nome do Caso de Uso	Encerrar Comanda
Autor(es) envolvidos	Administrador / Balcão / Cliente
Resumo	Ação de encerrar uma comanda em aberto
Pré-Condições	Estar autenticado como "administrador", usuário com perfil "Balcão" ou cliente
Pós-Condições	A comanda selecionada será encerrada
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Comanda". 2. O ator seleciona a comanda que deseja encerrar. 3. Selecionar a opção "Encerrar Comanda". 4. O status da comanda será alterado para fechada. 5. O sistema irá iniciar o caso de uso "Realizar Pagamento".
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Sair". 2. O sistema encerra a ação de encerrar a comanda. 3. O sistema retorna a tela principal do menu comanda.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O sistema não consegue encerrar a comanda solicitada. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.22: Caso de Uso - Pesquisar Comanda

Nome do Caso de Uso	Pesquisar Comanda
Autor(es) envolvidos	Administrador / Balcão
Resumo	Pesquisa uma comanda no sistema
Pré-Condições	Estar autenticado como "administrador" ou usuário com perfil "Balcão"
Pós-Condições	Exibe dados da comanda pesquisada
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Comanda". 2. O ator realiza a busca a através do código da comanda ou login do cliente nos campos de busca rápida ou filtro dinâmico.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Abrir Comanda". 2. O sistema abre o caso de uso para abrir a comanda.
Fluxo Alternativos A2	<ol style="list-style-type: none"> 1. O ator clica em "Encerrar Comanda". 2. O sistema abre o caso de uso para encerrar a comanda.
Fluxo Alternativos A3	<ol style="list-style-type: none"> 1. O ator clica em "Exportação". 2. O sistema abre o caso de uso para imprimir a comanda.
Fluxo Alternativos A4	<ol style="list-style-type: none"> 1. O ator clica em "Realizar Pedido". 2. O sistema abre o caso de uso para realizar pedido.
Fluxo Alternativos A5	<ol style="list-style-type: none"> 1. O ator clica em "Editar Pedido". 2. O sistema abre o caso de uso para editar pedido.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.23: Caso de Uso - Visualizar Comanda

Nome do Caso de Uso	Visualizar Comanda
Autor(es) envolvidos	Administrador / Balcão / Cliente
Resumo	Visualiza a sua comanda no sistema
Pré-Condições	Estar autenticado como "administrador", usuário com perfil "Balcão" ou cliente
Pós-Condições	Exibe todos os pedidos vinculados a comanda em aberto
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar no menu "Comanda". 2. O ator seleciona a comanda desejada através do filtro dinâmico. 3. Serão exibidos todos os pedidos realizado pelo cliente vinculados a sua última comanda em aberto.
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Abrir Comanda". 2. O sistema abre o caso de uso para abrir a comanda.
Fluxo Alternativos A2	<ol style="list-style-type: none"> 1. O ator clica em "Encerrar Comanda". 2. O sistema abre o caso de uso para encerrar a comanda.
Fluxo Alternativos A3	<ol style="list-style-type: none"> 1. O ator clica em "Exportação". 2. O sistema abre o caso de uso para imprimir a comanda.
Fluxo Alternativos A4	<ol style="list-style-type: none"> 1. O ator clica em "Realizar Pedido". 2. O sistema abre o caso de uso para realizar pedido.
Fluxo Alternativos A5	<ol style="list-style-type: none"> 1. O ator clica em "Editar Pedido". 2. O sistema abre o caso de uso para editar pedido.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.24: Caso de Uso - Imprimir Comanda

Nome do Caso de Uso	Imprimir Comanda
Autor(es) envolvidos	Administrador / Balcão
Resumo	Imprime ou envia por e-mail a comanda do cliente
Pré-Condições	Estar autenticado como "administrador"ou usuário com perfil "Balcão"
Pós-Condições	Cliente recebe a comanda impressa ou por e-mail
Fluxo Básico	<ol style="list-style-type: none"> 1. Clicar menu "Comanda". 2. O ator seleciona a comanda desejada através do filtro dinâmico. 3. Clicar no botão "Exportação"ou "Email". 4. Selecionar a opção "Imprimir"ou "PDF (Email)".
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

Tabela 3.25: Caso de Uso - Realizar Pagamento

Nome do Caso de Uso	Realizar Pagamento
Autor(es) envolvidos	Administrador / Balcão / Cliente
Resumo	Realiza o pagamento de uma comanda encerrada
Pré-Condições	Estar autenticado como "administrador", usuário com perfil "Balcão" ou cliente e a comanda desejada ter sido encerrada
Pós-Condições	O status da comanda é alterado para fechada
Fluxo Básico	<ol style="list-style-type: none"> 1. O ator seleciona o código da comanda através do filtro dinâmico. 2. O ator seleciona a comanda desejada e clica em "Encerrar Comanda". 3. O sistema irá direcionar para o PayPal para que o cliente realize o pagamento. 4. O sistema valida o pagamento. 5. O status da comanda é alterado para "FECHADA".
Fluxo Alternativos A1	<ol style="list-style-type: none"> 1. O ator clica em "Sair". 2. O sistema encerra a ação de pagar a comanda. 3. O sistema retorna a tela principal do menu comanda.
Fluxo de Exceções	<ol style="list-style-type: none"> 1. O ator não preenche os campos com valores válidos. 2. O sistema emite uma mensagem de erro. 3. O sistema retorna a anterior ao erro.

3.4.2 Modelo Entidade-Relacionamento

O MER presente na Figura 3.2 foi criado com o intuito de facilitar a criação do banco de dados na aplicação, com ele podemos visualizar rapidamente os elementos de cada tabela e o relacionamento entre elas.

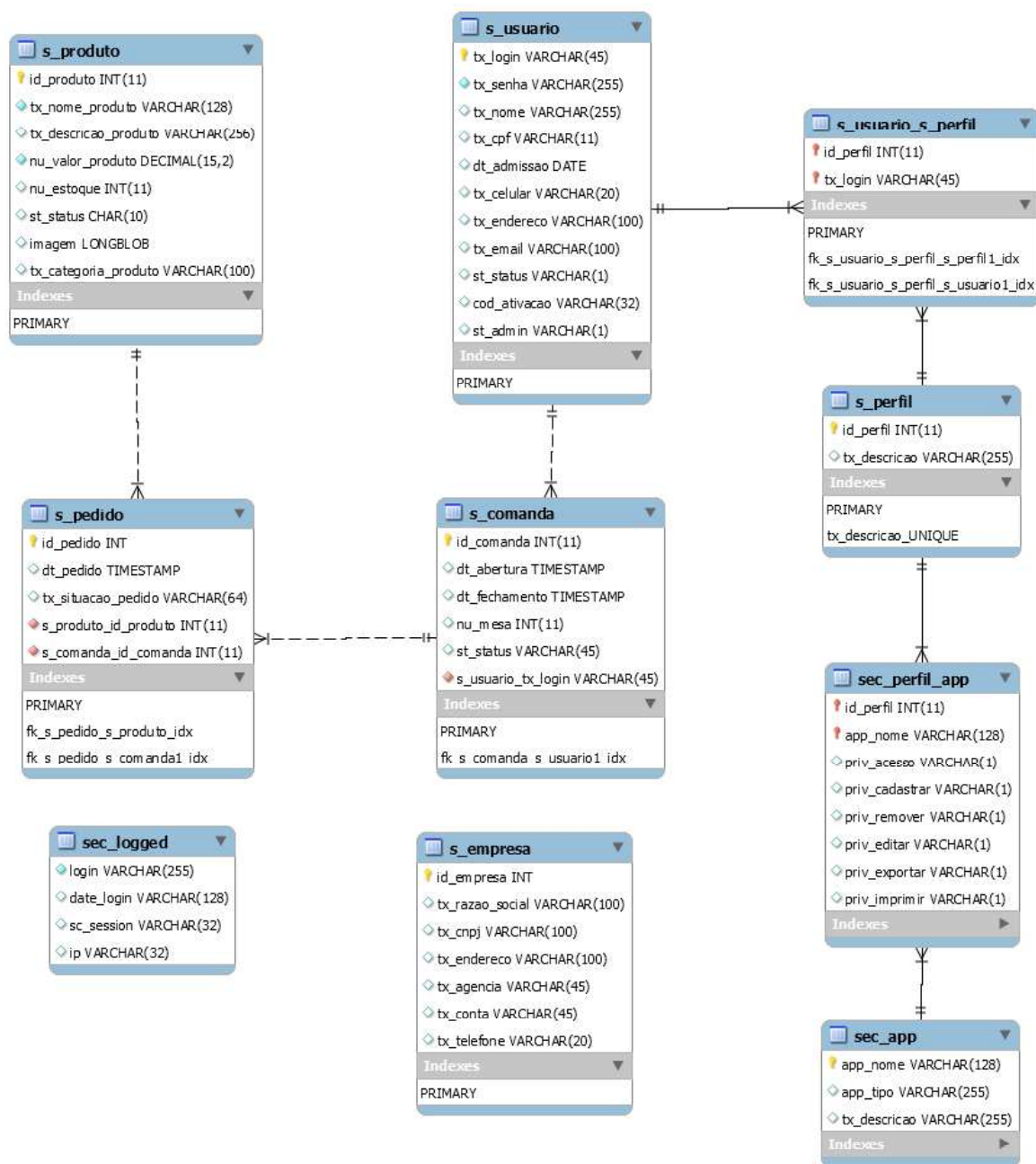


Figura 3.2: Modelo Entidade-Relacionamento

3.4.3 Dicionário de Dados

Tabela 3.26: Tabela s_produto

Campo	Tipo/Tamanho	Obrigatório	Comentário
id_produto	INT(11)	Sim	Chave primária do produto. Também chamada de código do produto no sistema.
tx_nome_produto	VARCHAR(128)	Sim	Armazena o nome do produto
tx_descricao_produto	VARCHAR(256)	Não	Armazena a descrição do produto
nu_valor_produto	DECIMAL(15,2)	Sim	Armazena o valor do produto
nu_estoque	INT(11)	Não	Armazena a quantidade do produto em estoque
st_status	CHAR(10)	Não	Indica se produto está disponível ou não
imagem	LONGBLOB	Não	Armazena a imagem do produto dentro do banco de dados
tx_categoria_produto	VARCHAR(100)	Não	Nome da categoria na qual o produto está relacionado

Tabela 3.27: Tabela s_pedido

Campo	Tipo/Tamanho	Obrigatório	Comentário
id_pedido	INT	Sim	Chave primária do produto. Também chamada de código do produto no sistema.
dt_pedido	TIMESTAMP	Não	Guarda a data e horário do pedido.
tx_situacao_pedido	VARCHAR(64)	Não	Indica se pedido foi entregue ou não.
s_produto_id_produto	INT(11)	Não	Chave Estrangeira
s_comanda_id_comanda	INT(11)	Não	Chave Estrangeira

Tabela 3.28: Tabela s_empresa

Campo	Tipo/Tamanho	Obrigatório	Comentário
id_empresa	INT	Sim	Chave primária do produto. Também chamada de código do produto no sistema.
tx_razao_social	VARCHAR(100)	Não	Armazena o nome da razão social da empresa.
tx_cnpj	VARCHAR(100)	Não	Armazena o CNPJ da empresa
tx_endereco	VARCHAR(100)	Não	Armazena o endereço físico da empresa
tx_agencia	VARCHAR(45)	Não	Armazena a agência referente aos dados bancários da empresa.
tx_conta	VARCHAR(45)	Não	Armazena o número da conta referente aos dados bancários da empresa.
tx_telefone	VARCHAR(20)	Não	Armazena a o número do telefone da empresa.

Tabela 3.29: Tabela s_usuario

Campo	Tipo/Tamanho	Obrigatório	Comentário
tx_login	VARCHAR(45)	Sim	Chave primária do produto. Também chamada de código do produto no sistema.
tx_senha	VARCHAR(255)	Sim	Armazena a senha do usuário.
tx_nome	VARCHAR(255)	Não	Armazena o nome do usuário.
tx_cpf	VARCHAR(11)	Não	Armazena o valor do CPF do usuário.
dt_admissão	DATE	Não	Armazena a data de admissão, utilizada para funcionários.
tx_celular	VARCHAR(20)	Não	Armazena número do celular do usuário.
tx_endereco	VARCHAR(100)	Não	Armazena o endereço residencial do usuário.
tx_email	VARCHAR(100)	Não	Armazena o email do usuário.
st_status	VARCHAR(1)	Não	Indica se usuário está ativo ou não.
cod_ativacao	VARCHAR(32)	Não	Código gerado para recuperação de senha.
st_admin	VARCHAR(1)	Não	Indica se usuário é administrador.

Tabela 3.30: Tabela s_comanda

Campo	Tipo/Tamanho	Obrigatório	Comentário
id_comanda	INT(11)	Sim	Chave primária armazena o código da comanda do cliente
dt_abertura	TIMESTAMP	Não	Armazena a data de criação da comanda
dt_fechamento	TIMESTAMP	Não	Armazena a data do encerramento da comanda.
nu_mesa	INT(11)	Não	Armazena o número da mesa ao qual a comanda está vinculada
st_status	VARCHAR(45)	Não	Armazena o status de 'ABERTA' ou 'FECHADA' da comanda
s_usuario_tx_login	VARCHAR(45)	Não	Chave estrangeira da tabela s_usuario

Tabela 3.31: Tabela s_usuario_s_perfil

Campo	Tipo/Tamanho	Obrigatório	Comentário
id_perfil	INT(11)	Sim	Chave primária e estrangeira da tabela s_perfil
tx_login	VARCHAR(45)	Sim	Chave primária e estrangeira da tabela s_usuario

Tabela 3.32: Tabela s_perfil

Campo	Tipo/Tamanho	Obrigatório	Comentário
id_perfil	INT(11)	Sim	Chave primária do perfil
tx_descricao	VARCHAR(255)	Não	Armazena a descrição do perfil

Tabela 3.33: Tabela sec_logged

Campo	Tipo/Tamanho	Obrigatório	Comentário
login	VARCHAR(255)	Sim	Chave primária da tabela. Armazena o login do usuário autenticado no sistema.
date_login	VARCHAR(128)	Não	Armazena a data do login realizado pelo usuário no sistema
sc_session	VARCHAR(32)	Não	Armazena o código da sessão do usuário
ip	VARCHAR(32)	Não	Armazena o IP do usuário autenticado

Tabela 3.34: Tabela sec_perfil_app

Campo	Tipo/Tamanho	Obrigatório	Comentário
id_perfil	INT(11)	Sim	Chave primária e estrangeira da tabela s_perfil
app_nome	VARCHAR(128)	Sim	Chave primária e estrangeira da tabela sec_app
priv_acesso	VARCHAR(1)	Não	Armazena o valor 'Y' para acesso e 'N' para negar acesso.
priv_cadastrar	VARCHAR(1)	Não	Armazena o valor 'Y' para permitir o cadastro e 'N' para negar cadastro.
priv_remover	VARCHAR(1)	Não	Armazena o valor 'Y' para permitir e 'N' para negar a remoção.
priv_editar	VARCHAR(1)	Não	Armazena o valor 'Y' para permitir e 'N' para negar a edição.
priv_exportar	VARCHAR(1)	Não	Armazena o valor 'Y' para permitir e 'N' para negar a exportação.
priv_imprimir	VARCHAR(1)	Não	Armazena o valor 'Y' para permitir e 'N' para negar a impressão.

Tabela 3.35: Tabela sec_app

Campo	Tipo/Tamanho	Obrigatório	Comentário
app_nome	VARCHAR(128)	Sim	Chave primária dos aplicativos.
app_tipo	VARCHAR(255)	Não	Armazena o tipo de aplicativo.
tx_descricao_produto	VARCHAR(255)	Não	Armazena a descrição do aplicativo

Capítulo 4

Desenvolvimento

4.1 Preparação do ambiente

Em sistemas Linux derivados do Debian a instalação de softwares é feita através de comando apt-get, no terminal. Para realizar as instalações deve-se ter permissões de administrador do sistema.

- Instalando Servidor Apache

O primeiro passo é a instalação do servidor HTTP Apache, com a linha "apt-get install apache2"

- Instalando MariaDB

Para instalar o MariaDB, utilizamos a linha de comando "apt-get install mariadb-server mariadb-client", em seguida configuramos a senha de root com a linha "mysql_secure_installation".

- Instalando PHP 7.0

Para instalar o PHP 7.0 basta digitar a linha de comando "apt-get install php7.0"no terminal.

Foram incluídos pacotes adicionais para estabelecer comunicação entre o php, o servidor apache e banco de dados com a linha "apt-get install libapache2-mod-php7.0 php7.0-mcrypt php7.0-mysql".

- Instalando PhpMyAdmin

Para instalar o phpMyadmin utilizamos a linha "apt-get install phpmyadmin php7.0-mbstring php-gettext".

foi necessário adicionar a linha "Include /etc/phpmyadmin/apache.conf"no arquivo apache2.conf

4.2 Criação do Banco de Dados

Como observado na Figura 3.2, foram necessárias 10 tabelas no banco de dados para a aplicação. Para elaboração do banco de dados e o relacionamento das tabelas, foi utilizada a ferramenta Mysql Workbench na versão 6.3. Esta ferramenta possibilita a modelagem do banco de forma rápida e fácil através da interface gráfica. Finalizada a modelagem, realizamos a exportação para o formato SQL. Para a administração do banco de dados foi utilizada a ferramenta phpMyAdmin.

A tabela *s_produto* guarda as informações dos produtos cadastrados pelo estabelecimento. Contendo os dados individuais de cada produto, como o nome, a descrição, o valor, a quantidade em estoque e o caminho da imagem do produto no servidor.

A tabela *s_comanda* listará os pedidos feitos pelo cliente, terá em seus dados as datas de abertura e fechamento, número da mesa e estará relacionada com o cliente.

A tabela *s_empresa* irá guardar as informações básicas do estabelecimento.

A tabela *s_pedido* guarda as informações dos pedidos realizados pelos clientes.

Na tabela *s_usuario* teremos os dados individuais de cada usuário que será vinculado a um perfil previamente criado. As informações necessárias para cadastrar um usuário são: login, senha, nome, cpf, número de celular, endereço e o grupo (perfil) ao qual esse usuário pertence.

A tabela *s_perfil* é responsável por armazenar os grupos que são usados pelo sistema. Nesta tabela irão ficar registrados os diferentes grupos que o estabelecimento precisa. Em nosso projeto, teremos, inicialmente, os grupos administrador, caixa, cliente, garçom e recepção.

A tabela *s_usuario_perfil* serve como tabela de apoio para a relação entre a tabela *s_usuario* e *s_perfil*.

As tabelas que começam com *sec_* serão responsáveis pela implementação de segurança ao sistema web. Dentre elas temos: *sec_app*, *sec_logged* e *sec_perfil_app*.

Na tabela *sec_app* ficarão as aplicações do sistema, como por exemplo, cadastrar produto ou abrir comanda. Essas aplicações serão utilizadas pela tabela *sec_perfil_app* para realizar um cruzamento de quais perfis podem acessar quais ações dentro do sistema.

Na tabela *sec_perfil_app* estão as permissões de que perfis possuem acesso a quais aplicações dentro do sistema web.

Na tabela *sec_logged* estão os dados de acesso ao sistema web. Essa tabela é usada para garantir que o mesmo usuário tenha somente um acesso, validando sua sessão e realizando o registro do acesso.

4.3 Desenvolvimento Aplicação Web

A aplicação web foi criada para prover o serviço de backend para o aplicativo Android e um sistema para os estabelecimentos realizarem o controle de seus cardápios, clientes, pedidos, comandas, pagamento e funcionários. No desenvolvimento do sistema web foi utilizado uma plataforma do tipo RAD, o scriptcase. As linguagens utilizadas no desenvolvimento foram PHP e HTML, JavaScript para funções lógicas, CSS para a estética das páginas e o banco de dados MySQL.

O scriptcase fornece uma interface amigável e de fácil interação para geração de códigos em PHP. Através da edição e seleção de parâmetros, via interface gráfica, é possível gerar funcionalidades que iriam exigir muito tempo e esforço.

Para melhorar a organização dos códigos do sistema, o ambiente foi dividido em seis diretórios: Cardápio, Comanda, Empresa, Menu, Segurança e _lib.

- **Cardápio**

Neste diretório estão os códigos PHP responsáveis exibir o cardápio e por realizar o CRUD dos produtos. O arquivo *consulta_cardapio.php* realiza a consulta do banco de dados trazendo todos os produtos cadastrados agrupados pela categoria. Além da consulta ao banco, esse código serve como tela principal da opção "Cardápio" do menu, sendo utilizada como ponto central para realizar as ações de CRUD de produtos.

- **Comanda**

Neste diretório estão os códigos PHP responsáveis pelo controle de comanda, pagamento e pedidos. Em seu arquivo principal, *crud_comanda.php*, está o código responsável por exibir as comandas em aberto, agrupadas pelo código da comanda, exibir os botões que realizam as ações de abrir comanda, encerrar comanda, editar pedido e realizar pedido.

- **Empresa**

Neste diretório está o código PHP responsável pelo cadastro e edição dos dados do estabelecimento. Através do código *form_empresa.php* é realizada a ação de inserir ou alterar os dados informados no formulário ao banco de dados.

- **Menu**

Neste diretório está o código PHP responsável pela barra de menu.

- **Segurança**

Neste diretório estão os códigos PHP responsáveis pela segurança e controle da aplicação. Todos os códigos desse diretório contém o prefixo "app_". Esse prefixo é uma devida a criação padrão do scriptcase. Dentre todos os códigos, os mais importantes

são: *app_Login.php*, *app_grid_sec_users.php* e *app_grid_sec_groups.php*. Os demais códigos servem de apoio para esses

O código *app_Login.php* é responsável pela página de login e controle da mesma. Não permitindo que usuários não autorizados acessem qualquer funcionalidade do sistema. O código *app_grid_sec_users.php* é responsável pelo controle de usuários do sistema web. E o código *app_grid_sec_groups.php* é responsável pelo controle dos perfis do sistema web.

- **_lib**

Neste diretório estão todas bibliotecas, imagens, extensões, arquivos css e códigos de apoio que auxiliam a execução do sistema web. Em destaque temos o subdiretório *prod* que contém os códigos utilizados pelo scriptcase para validação de valores, layout, framework, entre outros.

4.4 Desenvolvimento Aplicativo Android

O aplicativo Android tem como propósito ser o meio de acesso ao sistema por parte do cliente, nele é possível visualizar a comanda e fazer o pagamento.

Para seu desenvolvimento foi utilizado o react-native e seu código fonte é escrito inteiramente em JavaScript. Para que o react-native rode corretamente, foi necessário a instalação do Java, NodeJS e do Android Studio. A parte relacionada com o banco de dados é feita pelo sistema web com códigos em PHP.

O aplicativo consiste em 5 telas, cada uma criada como um componente a ser chamado pelo arquivo principal **App.js**. São elas:

- Login

A tela de Login exibe dois campos a serem preenchidos pelo cliente: email e senha para ter acesso. Caso o cliente não tenha cadastro, existe um botão que irá redirecioná-lo ao formulário de cadastro.

- Cadastro

A tela de cadastro tem campos para registrar o nome, cpf, email, número de celular e senha do cliente. Com esses dados é criado o registro do cliente, e o acesso é garantido para essas credenciais.

- Principal

Esta é a tela principal do aplicativo, nela existem botões para navegar entre todas as telas e funcionalidades. A navegação é feita utilizando o pacote react-navigation, para utilizá-lo é necessário adicioná-lo à pasta do projeto com o comando *npm install --save react-navigation*.

- Código

Esta tela exibe o código da comanda, é possível através desse código saber se o cliente tem acesso liberado para entrar no local e posteriormente se está liberado a deixar o local, este código também permite identificar o cliente mais rapidamente para fazer pedidos.

- Comanda

Esta tela exibe os itens consumidos pelo cliente dentro do estabelecimento, assim como os valores individuais e total. Há um botão que redireciona o cliente a realizar o pagamento utilizando um cartão de crédito e o mecanismo paypal.

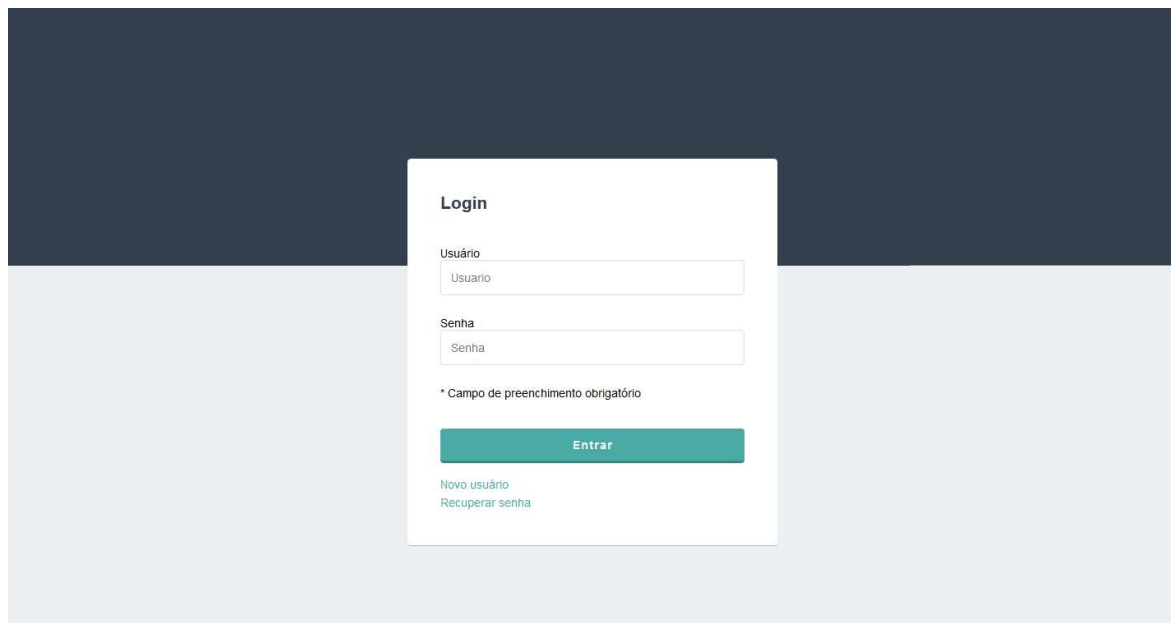
Capítulo 5

RESULTADOS E ANÁLISES

5.1 Sistema Web

O sistema foi hospedado no serviço contratado da Hostgator através do domínio, previamente, registrado no registro.br, byebyefila.com.br. O sistema está disponível através do endereço `http://byebyefila.com.br`.

Ao acessar a página web, será exibida uma tela de login onde o usuário administrador do estabelecimento poderá realizar o primeiro acesso para começar as configurações do estabelecimento dentro do sistema. A página de login está apresentada na figura abaixo.



Login

Usuário

Senha

* Campo de preenchimento obrigatório

[Entrar](#)

[Novo usuário](#)
[Recuperar senha](#)

Figura 5.1: Página Inicial - Login

Após o login, é exibida uma tela contendo seis opções no menu. Esse menu é fixo e ao selecionar uma das opções oferecidas por ele, será aberta o resultado no painel central do sistema web. Na Figura (4.2) é possível ver o layout da página após o login.

Figura 5.2: Página após o login

Uma das primeiras configurações necessárias pelo administrador é a de cadastrar as informações do seu estabelecimento clicando na opção **Empresa** do menu e preenchendo as informações de razão social, CNPJ, endereço, agência, conta e telefone do estabelecimento. Finalizando o preenchimento dos campos solicitados basta clicar em *Salvar* para realizar o registro das informações no banco. A página de cadastro e edição dos dados da empresa podem ser visualizados na Figura 4.3.

DADOS DA EMPRESA

Salvar

Razao Social	Night Club
CNPJ	25.986/0036-15
Endereço	Endereço do Estabelecimento
Agência	35688-9
Conta	124357-6
Telefone	613305689

Figura 5.3: Empresa

O controle do cardápio e dos produtos foram juntados na opção **Cardápio** do menu. Nesta página é possível cadastrar, editar, pesquisar e excluir os produtos que estarão no cardápio. Uma funcionalidade interessante é a possibilidade de imprimir o cardápio conforme

exibição na tela, ou seja, é possível que o estabelecimento filtre uma categoria de produto e imprima apenas tal categoria. O layout da opção **Cardápio** pode ser visto na figura abaixo.

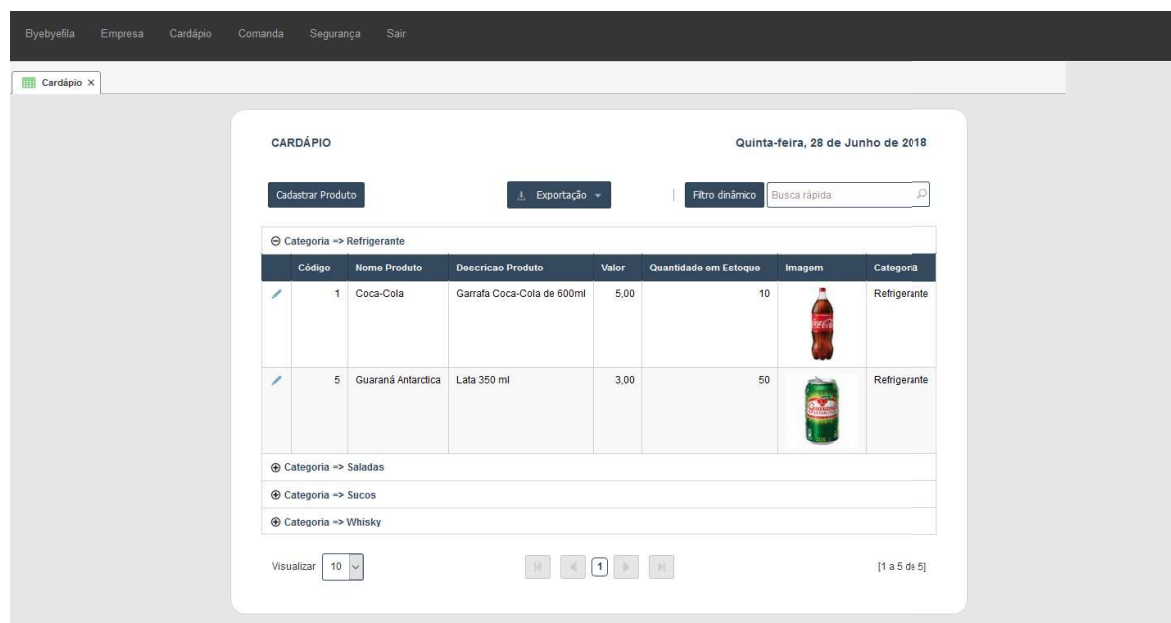


Figura 5.4: Controle de cardápio e produto

A opção **Comanda** no menu é responsável pelo controle da comanda digital. Nesta tela é possível abrir uma nova comanda, encerrar uma comanda, realizar um pedido, editar um pedido e visualizar as comandas do estabelecimento. A visualização dessa página pode ser vista na Figura 4.5.

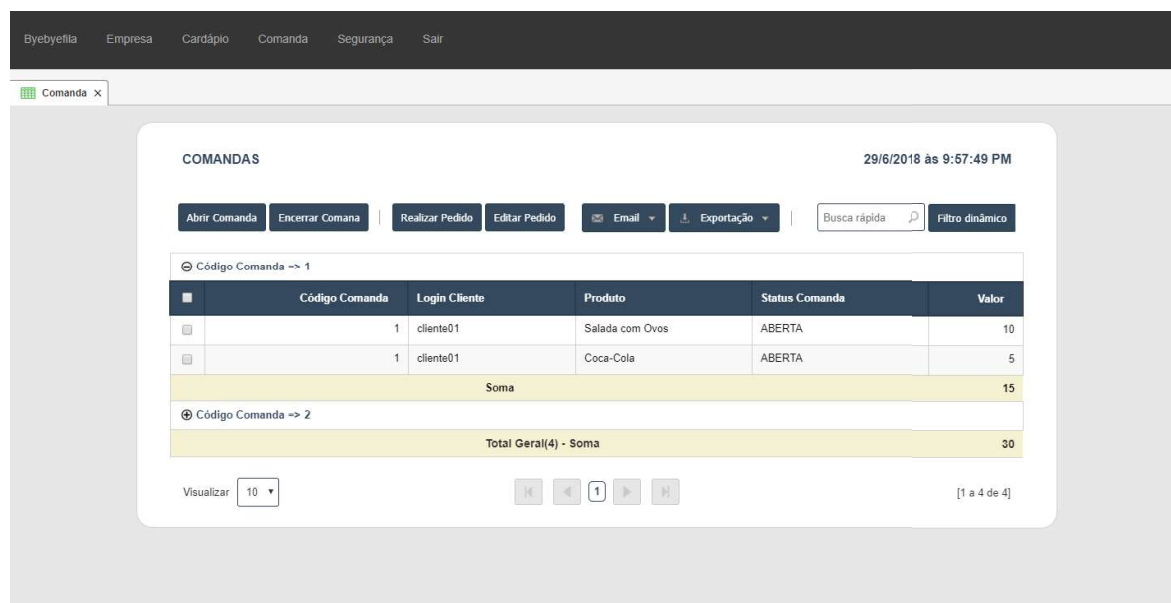


Figura 5.5: Comanda

Na Figura 4.6 é possível ver a tela de confirmação quando o funcionário realiza o encerramento da comanda. Ao se confirmar, clicando no botão **PayPal**, ocorre a chamada da API

do PayPal para realização do pagamento das comandas e produtos selecionados na Figura 4.5.



Figura 5.6: Comanda

Por fim, todo o controle de usuários, perfis e controle de acesso é realizado através da opção **Segurança** disponível no menu. Conforme pode ser visto na Figura abaixo, essa opção possui várias subopções.

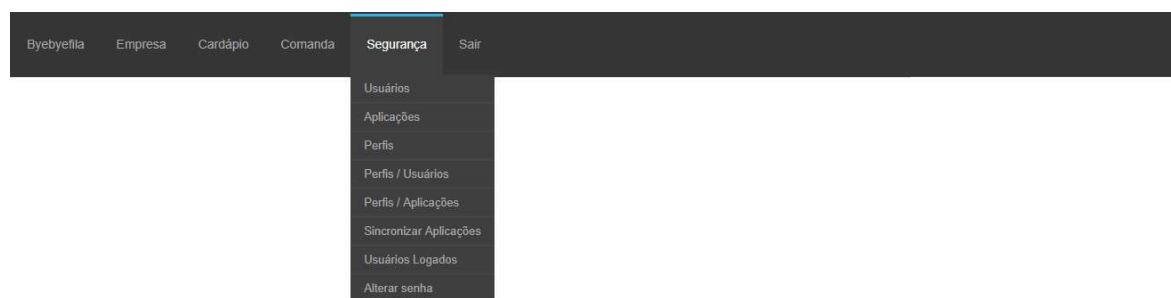


Figura 5.7: Segurança

O controle de usuários, perfis e acesso foi sub dividido pensando em facilitar o gerenciamento por parte do administrador do estabelecimento. A funcionalidade de cada subopção pode ser vista abaixo.

- **Usuários**

Nesta subopção é possível realizar o CRUD dos usuários ao sistema.

- Aplicações

Nesta subopção o administrador pode incluir ou editar uma aplicação ao sistema web. Ao adicionar uma aplicação, será possível gerenciar que perfis podem acessar tal aplicação.

- Perfis

Nesta subopção é possível administrar os perfis/grupos do sistema web. Por padrão, são criados cinco grupos: Administrador, caixa, cliente, garçom e recepção.

- Perfis / Usuários

Nesta subopção é realizada a inclusão dos usuários ao grupo/perfil desejado. Por padrão, um novo usuário é incluído ao grupo/perfil de Cliente.

- Perfis / Aplicações

Nesta subopção é possível realizar o controle de acesso dos grupos/perfis as aplicações desejadas. Através dessa opção podemos definir que o grupo/perfil garçom pode apenas realizar pedido, por exemplo.

- Sincronizar Aplicações

Esta opção foi incluída pensando em novas aplicações criadas futuramente. Dessa forma, é possível incluir outra aplicação ao controle de acesso através dessa funcionalidade.

- Usuários Logados

Essa funcionalidade foi criada para fins de controle de quem está logado no sistema. Nesta opção é possível ver que usuário está logado no sistema, seu IP e a data do login.

- Alterar Senha

Esta subopção fornece a funcionalidade de alterar a senha do usuário.

5.2 Aplicativo Android

A primeira tela do aplicativo é a de Login, após entrar com as credenciais o cliente terá acesso ao sistema. Estão presentes dois campos, um para o email do usuário e outro para senha pré-cadastrada respectivamente. Para novo clientes, existe um link para fazer novos cadastros.

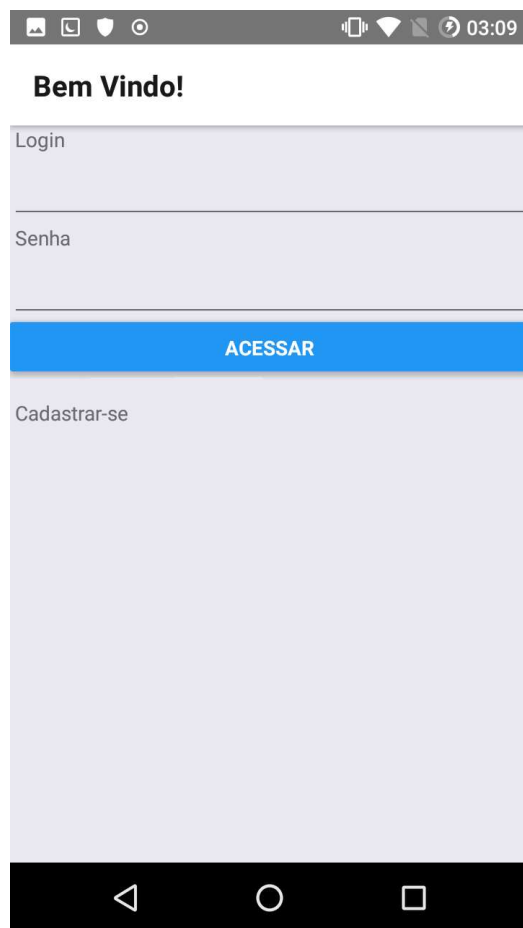


Figura 5.8: Tela de Login do Aplicativo

Na tela de cadastro o usuário deve preencher cinco campos, nome completo, número do CPF, e-mail, número do celular, e senha respectivamente. Após o cadastro o usuário pode efetuar o login e ter acesso ao sistema.

The screenshot shows a mobile application interface for user registration. At the top, there is a status bar with icons for signal, Wi-Fi, battery, and the time 03:09. Below the status bar is a navigation bar with a back arrow and the title 'Cadastro de Usuario'. The main area contains six text input fields, each with a label: 'Nome Completo:', 'CPF:', 'e-mail:', 'Numero de celular:', 'Senha:', and 'Confirmar Senha:'. Below these fields is a blue button with the text 'ENVIAR CADASTRO'. At the bottom of the screen is an Android-style navigation bar with three icons: a back arrow, a circle, and a square.

Figura 5.9: Campos para o cadastro de cliente

Ao fazer o login o usuário tem acesso ao menu principal, contendo três botões para navegação entre telas, os botões são: Código da Comanda, Comanda e Log Out.

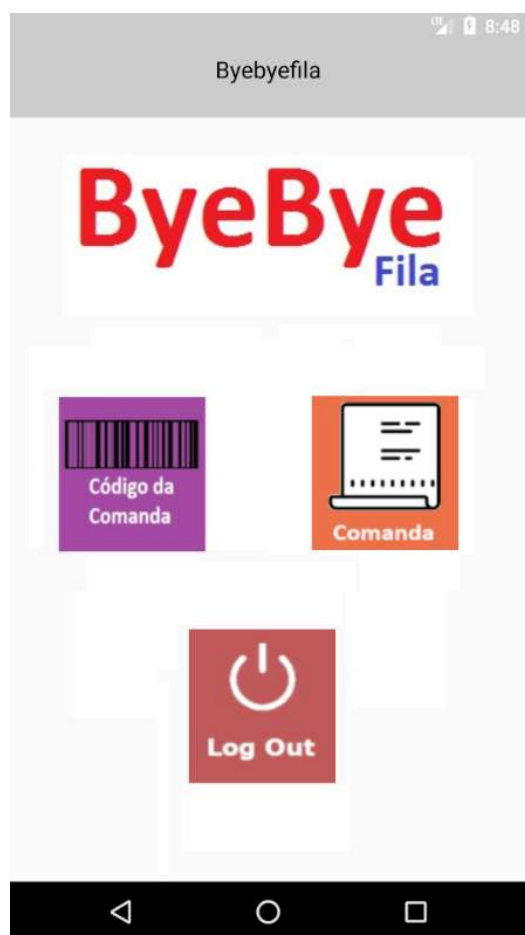


Figura 5.10: Menu Principal

A primeira tela exibe o código da comanda, com ele é possível adicionar novos itens à comanda e consultar o status da mesma, assim possibilitando a liberação da entrada ou saída do estabelecimento.



Figura 5.11: Código da comanda

A segunda tela exibe os itens consumidos pelo cliente e o valor total da conta. através dessa tela o cliente tem controle do que está sendo gasto, assim como pode conferir a qualquer momento o que está sendo colocado em sua comanda. Logo abaixo existe um botão que direcionará o cliente ao mecanismo de pagamento paypal.



Figura 5.12: Comanda eletrônica

Por fim o Botão de Log Out fará com que o cliente encerre sua seção no aplicativo, após pressioná-lo o usuário retorna à tela de login.

Capítulo 6

CONCLUSÃO E TRABALHOS FUTUROS

Este projeto foi implementado com o objetivo de solucionar um problema que percebemos se tornar recorrente no dia a dia. Com a implementação do sistema Byebyefila os estabelecimentos e clientes poderão tirar proveito da tecnologia para otimizar seu tempo. A plataforma web desenvolvida irá auxiliar os bares, restaurantes e casas noturnas a utilizarem os smartphones como dispositivo a auxiliar a integração do negócio com os clientes.

É importante frisar que o projeto foi desenvolvido pensando em sanar um problema enfrentado pelos autores deste projeto durante um encontro em uma boate, mas isso não exclui a possibilidade desse sistema ser utilizado em outros estabelecimentos como bares e restaurantes. Da forma como esse sistema foi implementado, é possível sua expansão e contínuo desenvolvimento.

A metodologia e o conhecimento usados para desenvolver esse projeto foram baseados nas diversas disciplinas cursadas pelos autores deste projeto ao longo do curso de Engenharia de Redes de Comunicação.

Com intuito de aperfeiçoar este projeto notamos que podem ser implementados os seguintes itens como trabalhos futuros.

- Integração do aplicativo Android com sistema de terceiros.
- Inclusão de módulo para nota fiscal.
- Inclusão de módulo para gerenciamento financeiro do estabelecimento.
- Implementação de suporte a leitoras externas, como leitor de cartões, código de barras e QR-Code.
- Aperfeiçoamento do design do sistema web e aplicativo Android.
- Adicionar funcionalidade de realizar pedidos através do aplicativo Android.

- Melhorar a segurança do cadastro pelo aplicativo Android.

Referências Bibliográficas

- [1] Alan Lakein. *How to Get Control of Your Time and Your Life*. New American Library, New York City; ISBN 0-451-13430-3, 1973.
- [2] Alan Lakein. *Give Me a Moment and I'll Change Your Life: Tools for Moment Management*. Andrews & McMeel, Kansas City, MO; ISBN 0-8362-3591-6, 1997.
- [3] The Statistics Portal - Statistics and Studies. Disponível em:
<https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>
- [4] SOAWebServices - Connecting Systems. Disponível em:
<http://www.soawebservices.com.br/como-funciona.aspx>
- [5] O que é LAMP? Disponível em:
<https://www.hardware.com.br/dicas/intro-lamp.html>
- [6] O que é um SGBD? Disponível em:
<https://www.oficinadanet.com.br/post/16631-o-que-e-um-sgbd>
- [7] O que é Kernel?. Disponível em:
<https://www.oficinadanet.com.br/post/13858-o-que-e-kernel>
- [8] Netcraft - Web Server Survey. Disponível em:
http://news.netcraft.com/archives/web_server_survey.html
- [9] Conceitos Fundamentais de Banco de Dados. Disponível em:
<https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>
- [10] Sobre o Mariadb. Disponível em:
<https://mariadb.com/kb/pt-br/sobre-o-mariadb/>
- [11] MariaDB. Disponível em:
<https://mariadb.com/>
- [12] O que é um SGBD?. Disponível em:
<https://dicasdeprogramacao.com.br/o-que-e-um-sgbd/>

- [13] SGBD – Sistema de Gerenciamento de Banco de Dados. Disponível em:
<http://www.diegomacedo.com.br/sghd-sistema-de-gerenciamento-de-banco-de-dados/>
- [14] phpMyAdmin. Disponível em:
<https://phpmyadmin.readthedocs.io/en/latest/faq.html>
- [15] O que é JavaScript?. Disponível em:
<http://tableless.github.io/iniciantes/manual/js/>
- [16] JavaScript Tutorial. Disponível em:
<https://www.w3schools.com/js/default.asp>
- [17] Tutorialspoint. *Android application development*. Disponível em:
http://www.tutorialspoint.com/android/android_tutorial.pdf
- [18] Peter Pin-shan Chen . *Paper:The Entity-Relationship Model: Toward a Unified View of Data (1976)* . Disponível em:
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.123.1085>
- [19] Booch G. *The Unified Modeling Language User Guide*. Addison-Wesley Professional, 2005.
- [20] Roger S. Pressman, Burce R. Maxim. *Appendix 1 An Introduction to UML in book Software Engineering: A Practitioner's Approach 8 Ed.*
- [21] Diagrama de Casos de Uso. Disponível em:
<http://www.dsc.ufcg.edu.br/~sampaio/cursos/2007.1/Graduacao/SI-II/Uml/diagramas/usecases/usecases.htm>
- [22] Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER). Disponível em: <https://www.devmedia.com.br/modelo-entidade-relacionamento-e-diagrama-entidade-relacionamento-der/14332>
- [23] 4 - Modelo Entidade e Relacionamentos. Disponível em:
<https://sites.google.com/site/uniplibancodedados1/aulas/aula-4--modelo-entidade-e-relacionamentos>
- [24] IDC - Analuse the Future. Disponível em:
<https://www.idc.com/promo/smartphone-market-share/os>
- [25] O que é Kernel? Disponível em:
<https://www.tecmundo.com.br/macros/1636-o-que-e-kernel-.htm>
- [26] A história do Linux. Disponível em:
<https://www.tecmundo.com.br/sistema-operacional/4228-a-historia-do-linux.htm>

- [27] O que é PHP? Para que serve?. Disponível em:
<http://phpdozeroaoprofissional.net.br/o-que-e-php-para-que-serve/>
- [28] O que é o PHP?. Disponível em:
https://secure.php.net/manual/pt_BR/intro-what-is.php
- [29] O que o PHP pode fazer?. Disponível em:
https://secure.php.net/manual/pt_BR/intro-what-cando.php
- [30] CSS Introduction. Disponível em:
https://www.w3schools.com/css/css_intro.asp
- [31] Ubuntu 16.04 (LTS) LAMP server tutorial with Apache, PHP 7, and MySQL. Disponível em:
<https://www.howtoforge.com/tutorial/install-apache-with-php-and-mysql-on-ubuntu-16-04-lamp/>
- [32] Understanding LAMP. Disponível em:
<https://www.serverwatch.com/tutorials/article.php/3567741/Understanding-LAMP.htm>
- [33] Web Server Working Diagram. Disponível em:
<https://image.slidesharecdn.com/modperlwithapache-151221180023/95/building-dynamic-websites-with-mod-perl-and-apache-13-638.jpg?cb=1450721033>
- [34] With knowledge in HTML, CSS and a little JavaScript, what kind of projects should I start with to strengthen my skills in web development? Disponível em:
<https://www.quora.com/With-knowledge-in-HTML-CSS-and-a-little-JavaScript-what-kind-of-projects-should-I-start-with-to-strengthen-my-skills-in-web-development>
- [35] PHP e MySQL. Disponível em:
http://www.plus2net.com/php_tutorial/php-mysql.php
- [36] Why we use React Native for mobile app development. Disponível em:
<http://altitudelabs.com/blog/react-native-mobile-app-development/>
- [37] SGDB (Sistema de Gestão de Bases de Dados). Disponível em:
<https://sisinfosite.wordpress.com/2015/10/16/sgdb-sistema-de-gestao-de-bases-de-dados/>
- [38] THE WORLD'S LARGEST WEB DEVELOPER SITE. Disponível em:
<https://www.w3schools.com/>